Samsung's Use of Apple Patents in Smartphones

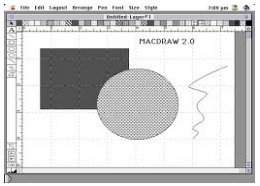
Key issues

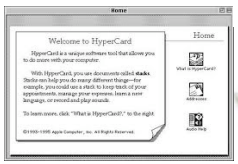
- Samsung is a key Apple partner
- The Android software platform makes extensive use of Apple intellectual property ... without Apple's permission
    - Android is designed to lead companies to imitate the iPhone product design and strategy
- Samsung's choice to use Android without a license undermines Samsung's greater relationship with Apple

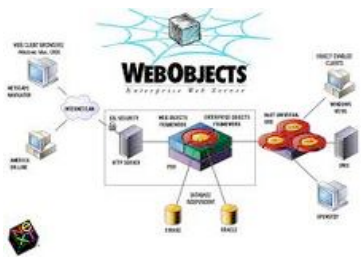
Apple Patents

- Large worldwide patent portfolio on computing technology
  - Over 3500 US patents
  - Worldwide coverage for many key patents
- Tracks Apple's 35 years of leadership in personal and mobile computing and communications
- Definitive patent portfolio for industry --
  - Modern computing and consumer electronics hardware
  - Internet architecture and services
  - Modern software - OS, applications, and user interface

Apple: 35 Years of Market Leading Innovation







*Confidential * Provided for Information and Business Settlement Purposes Only*
Information and Disclosure Provided Under Rule 408 of Federal Rules of Evidence, Without Prejudice

Apple Patents: General Computing

- Apple patents broadly cover general computing technologies used in all modern computing devices and consumer electronics
  - Core processor technologies
  - High speed internal buses and peripheral device buses
  - Graphics processors
  - Networking and communications
  - Power management
  - Mechanical and input technologies

Apple Patents: Internet architecture and services

- Apple patents are essential to modern Internet-enabled devices and services
  - Dynamic web page generation
  - Persistent objects in a web page
  - Object/ relational database mapping
  - Location based services
  - Mobile computer IP address assignment
  - Media and application store services
  - Multimedia format and delivery

Apple Patents: Software

- Apple has definitive patents on modern operating systems and applications architecture
  - Operating systems and basic software architecture
  - Graphics
  - User Interface
  - Multimedia processing
  - Networking and communication
  - Object oriented software
  - Development technologies

Convergence In Modern Smartphones

Apple core computing

Apple core computing technologies for modern and mobile computing developed over 20 years on desktop and laptop computer platform

- Modern, real computing platform
- Extensible, programmable OS
- Software application platform
- Advanced, friendly user interface
- Graphics and multimedia
- Component-based architecture with component links
- Internet and network technology

Basic telephony

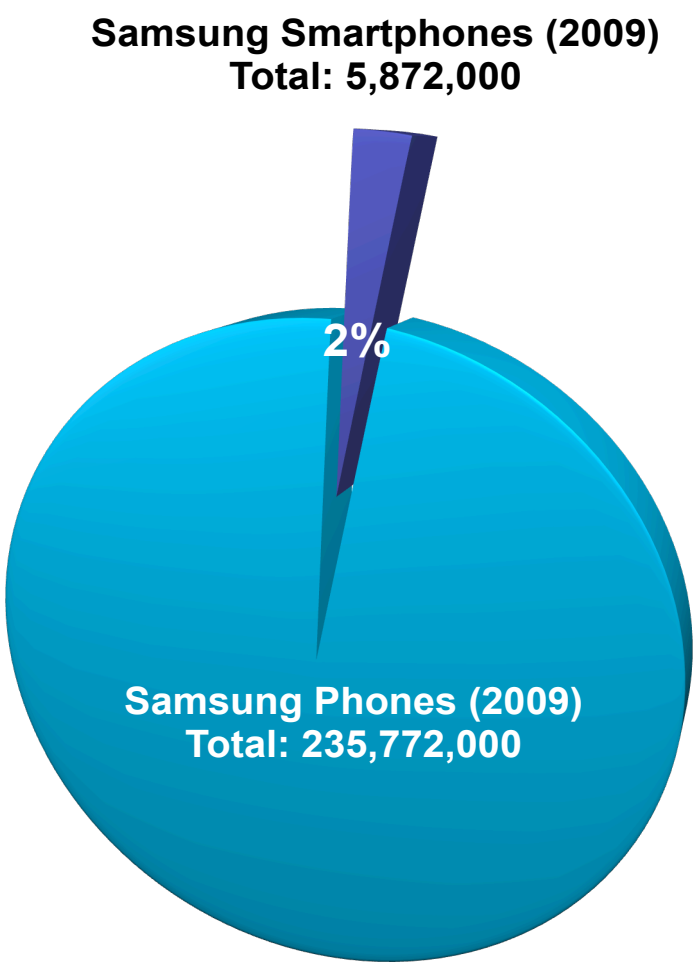
Basic network interface -- industry-standardized air interface (least common denominator) and data format

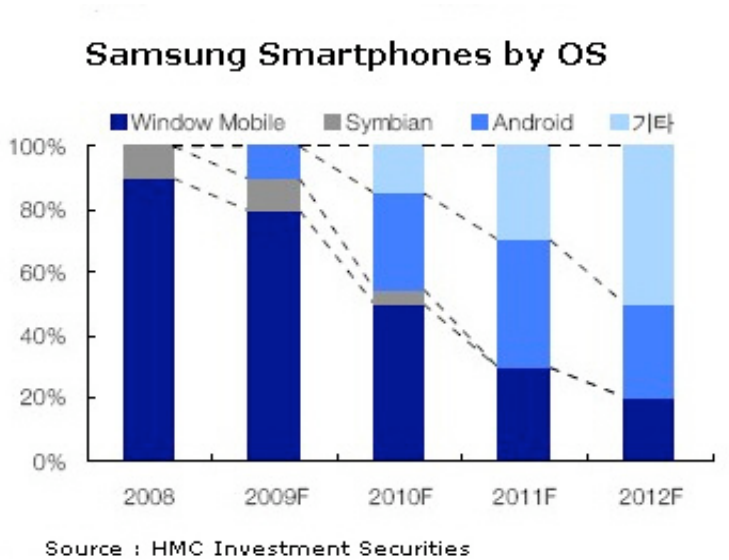
Apple advanced features

Apple's iPod and iPhone innovations have defined the standard for modern high-end consumer devices

- Multitouch user interface
- Apps and App Store
- iTunes media store and media player
- Real Web and Web services
- Advanced sensors and device context
- Service-oriented offering

Samsung Smartphone Sales



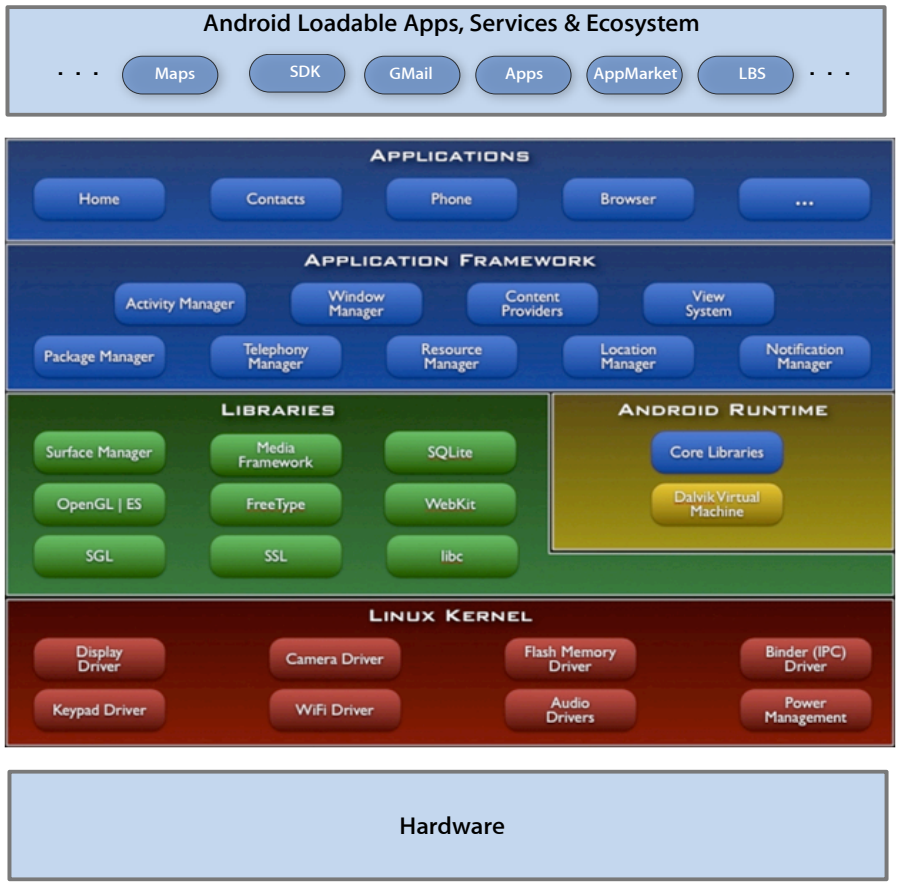


Source : HMC Investment Securities

Apple Patents and Android

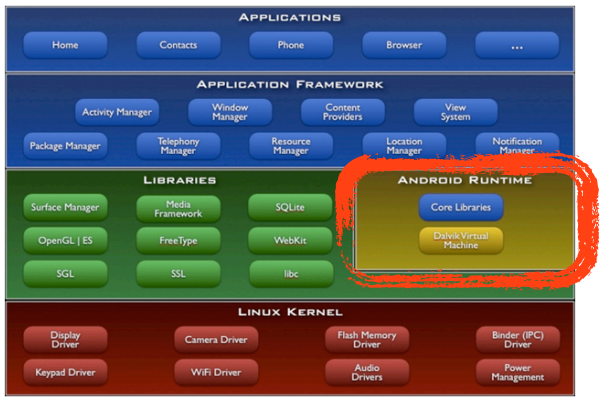
- Apple has identified dozens of examples where Android is using or encouraging others to use Apple patented technology
- Many more Apple patents are relevant to the Android platform
- Apple has not authorized the use of any of these patents
- Limited examples to follow

Android Architecture



Apple Patents Infringed by Android Runtime

Example Apple patents infringed by Samsung phones using Android runtime

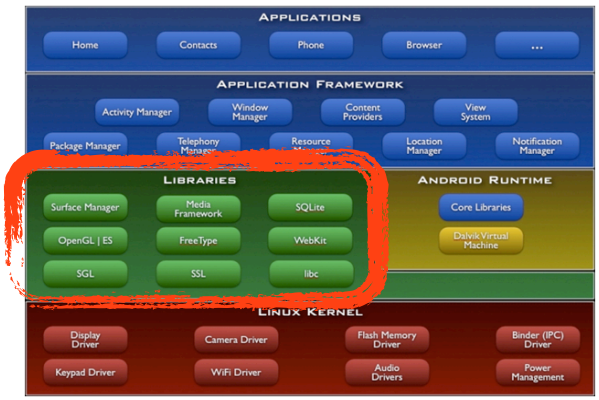


- U.S. 6,424,354 (EP, JP)* - Interest-based notification
- U.S. 5,481,721* - Dynamic binder
- U.S. 5,519,867*- Object oriented multitasking
- U.S. 6,275,983 (EP, JP, CN)* - Wrapper loader
- U.S. 5,367,633 (EP, JP) - OO notification framework
- U.S. 5,566,337 (EP, JP)* - Event handling
- U.S. 5,915,131 (EP)* - Tailored distinct IO APIs
- U.S. 5,969,705* - Background event handling
- U.S. 6,684,261 (EP, JP, CN) - OO operating system
- U.S. 5,379,432 (EP, JP, CN) - Wrappers
- U.S. 7,380,116 (JP) - Real-time display adaptation
- U.S. 6,067,577 - Dynamic resolution
- U.S. 5,911,067 - Application switching
- U.S. 5,911,069 - Exception handling
- U.S. 5,404,529 - IPC Wrapper
- U.S. 5,473,777 - VM Wrapper
- U.S. 5,475,845 (EP, JP, CN) - Wrapper system

* Asserted against HTC in Pending Litigations

Apple Patents Infringed by Android Libraries

Example Apple patents infringed by Samsung phones using Android libraries

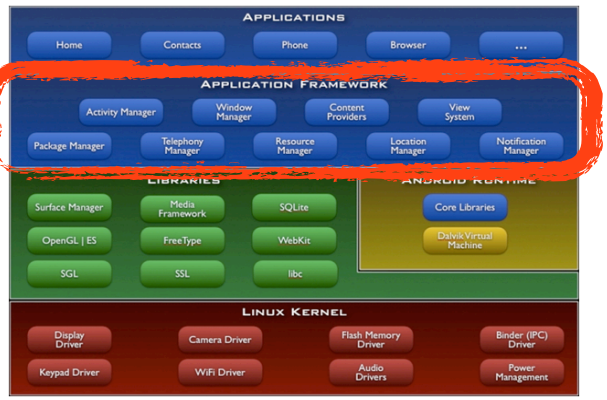


- U.S. 6,343,263 (EP)* - Realtime signal processing APIs
- U.S. 5,920,726* - OS camera management
- U.S. 7,281,212 - Multi-track media
- U.S. 7,043,694 - Multi-track media
- U.S. 5,379,129 - Compositing images
- U.S. 5,404,447 (GB) - Manipulating pixel streams
- U.S. 7,292,636 (EP,CN,JP) - Processing a video picture
- U.S. 6,757,438 - Video compression
- U.S. 6,728,315 (EP,CN,JP) - Digital video encoding
- U.S. 5,828,904 - Synchronizing data retrieval
- U.S. 6,098,126 - Synchronizing data retrieval

* Asserted against HTC in Pending Litigations

Apple Patents Infringed by Android Application Framework

Example Apple patents infringed by Samsung phones using Android application framework



- U.S. 5,455,599 (EP, JP)* - OO graphics framework
- U.S. 7,362,331 (EP, JP)* - Non-linear animation of GUI
- U.S. 6,031,532 - Composite images
- U.S. 5,929,852 (EP)* - Network widget
- U.S. 6,344,855 (EP) - Encapsulated entity
- U.S. RE39,486 (EP)*- Extensible NW component system
- U.S. 7,469,381* - Scrolling with bounce and snapback
- U.S. 6,259,446 (EP, JP) - Menu system
- U.S. 6,593,947 - OO image rendering
- U.S. 6,956,564 (EP, CN) - Tilt-based display mode
- U.S. 5,764,218 - Touch gesture control
- U.S. 5,469,194 - Orientation-aware touch interface
- U.S. RE41,088 - Orientation of captured image
- U.S. 6,282,646 - Adaptive display configuration
- U.S. 7,003,260 - Database programs for handhelds
- U.S. 5,455,854 (EP) - Telephony system
- U.S. 7,084,859 - Tactile touch screen
- U.S. 5,880,729 (EP, JP) - Animated transitions
- U.S. 5,196,838 - Autoscrolling

* Asserted against HTC in Pending Litigations

Apple Patents Infringed by Android Applications Layer

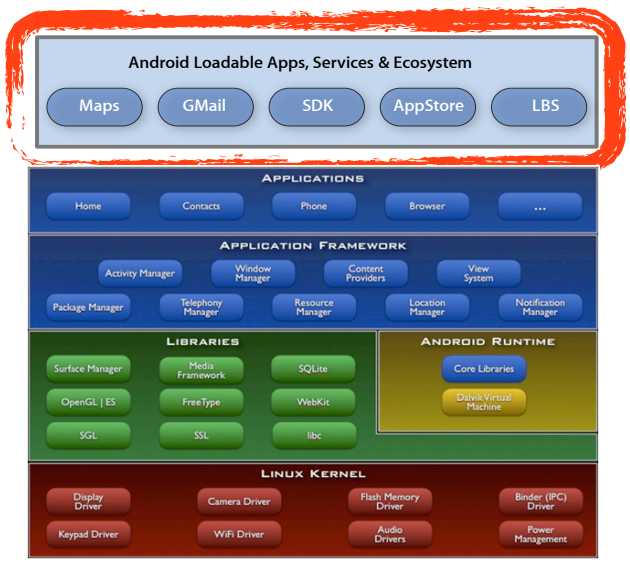
Example Apple patents infringed by Samsung phones using Android standard applications

- U.S. 7,657,849 (EP, JP)* - Swipe to unlock
- U.S. 5,946,647* - Data detectors
- U.S. 7,479,949*- Multiple distinct touch heuristics
- U.S. 7,602,378 - Selective soft keypad
- U.S. 5,128,672 (JP) - Dynamic predictive keyboard
- U.S. 6,236,396 - Calendar data entry
- U.S. 7,479,971 - Automatic window scrolling
- U.S. 5,612,719 - Gesture sensitive buttons
- U.S. 7,669,134 - Messaging UI
- U.S. 6,072,489 - Translucent user interfaces
- U.S. 5,949,432 - Translucent user interfaces
- U.S. 5,544,358 - Card/ list view for contacts
- U.S. 5,446,882 - Card/ list views for contacts
- U.S. 5,603,053 - Pop-up interactive tools
- U.S. 6,493,002 - Status bar

* Asserted against HTC in Pending Litigations

Apple Patents Infringed by Android Ecosystem

Example Apple patents infringed by Android phones offering Android loadable apps, Internet services and ecosystem



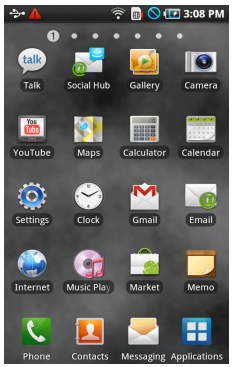
- U.S. 7,421,690 - Threaded email
- U.S. 5,926,190 - Virtual reality images
- U.S. 7,187,997 - LBS
- U.S. 7,710,290 - Invocable speed reference
- U.S. 7,003,260 - Database programs for handheld
- U.S. 5,555,369 - SDK with device emulator
- U.S. 5,572,582 (EP) - Teleconference communication
- U.S. 6,910,052 - Software update
- U.S. 6,430,576 - Software update
- U.S. 7,584,468 - Software update
- U.S. 7,660,831 (EP) - Data synchronization
- U.S. 5,710,922 - Last-time altered sync
- U.S. 6,253,228 (EP) - Package-based sync
- U.S. 6,947,967 (EP) - Package-based sync

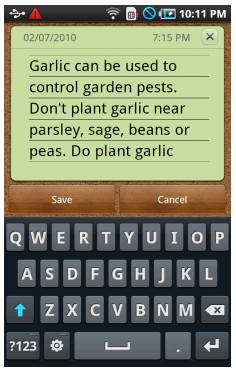
Samsung Copying iPhone

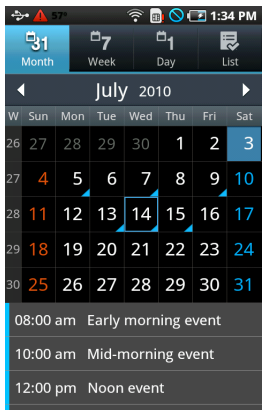
Apple iPhone 4

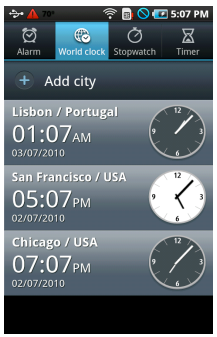
Samsung Galaxy S

Samsung Copying iPhone









Apple iPhone 4

Samsung Galaxy S

Samsung Copying iPhone

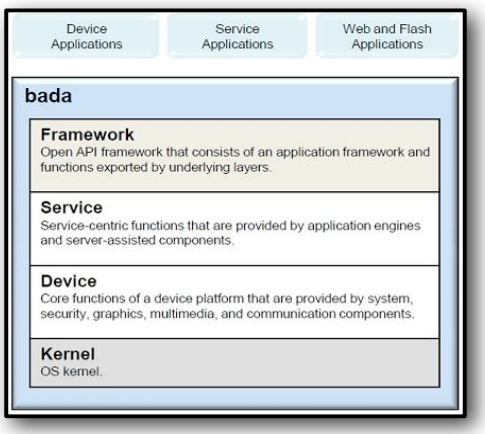
Apple iPhone 4

Samsung Galaxy S

Macworld 2007 - January 9 ,2007

Android Phones

Samsung's Bada Platform



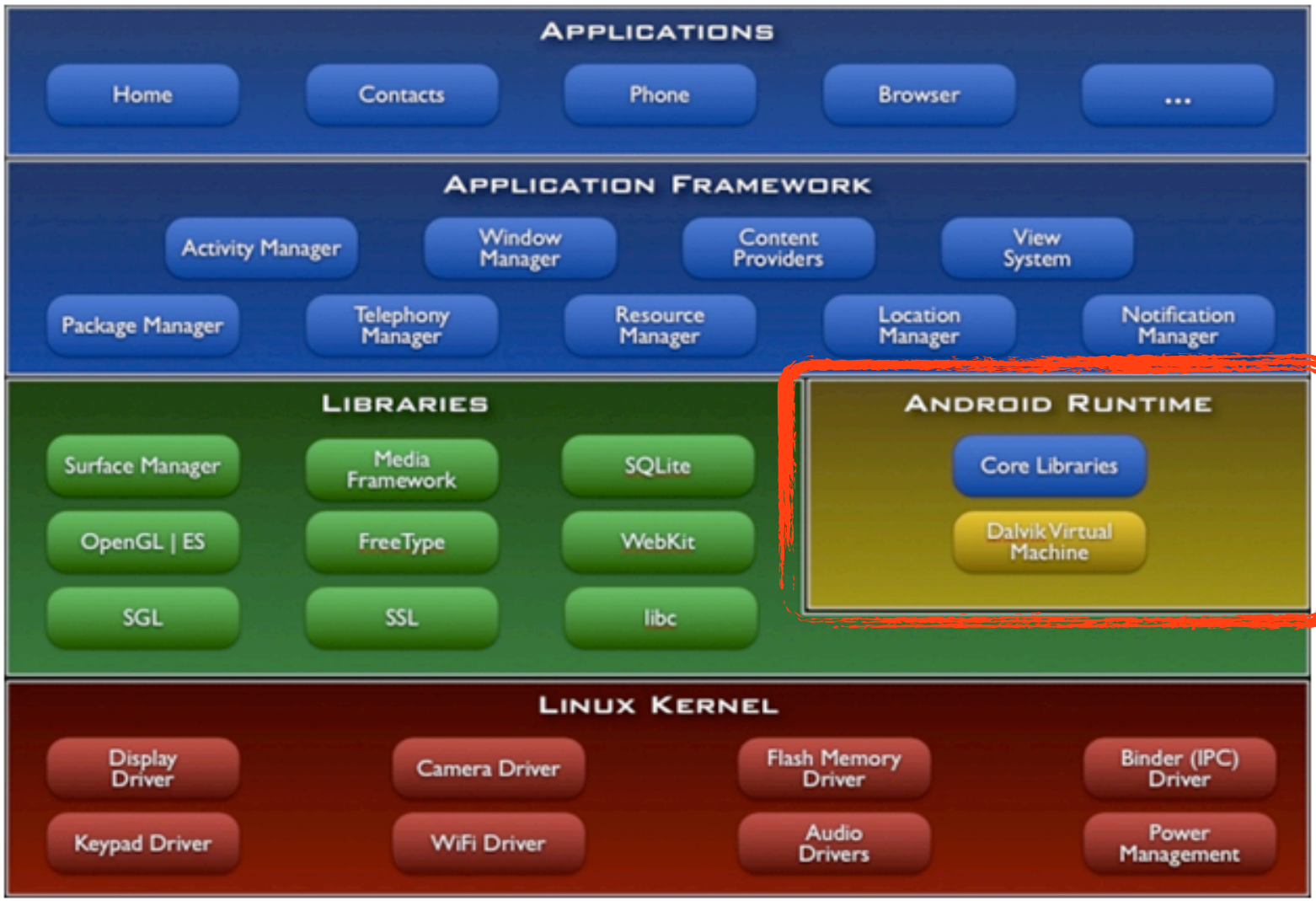
TOUCHWiZ

Key issues

- Samsung is a key Apple partner
- The Android platform is causing Samsung to unfairly use Apple's intellectual property to undermine and imitate iPhone
  - Symbian and Bada also using Apple patents
- Samsung needs a license to continue to use Apple patents in infringing smartphones

Example Patents Infringed by All Samsung Android Phones

Android Stack



Highlights of Example Apple Patents Infringed by Android Runtime

U.S. 6,424,354 & 5,367,633 (Interest-Based Notification)

U.S. 5,481,721 (Binder)

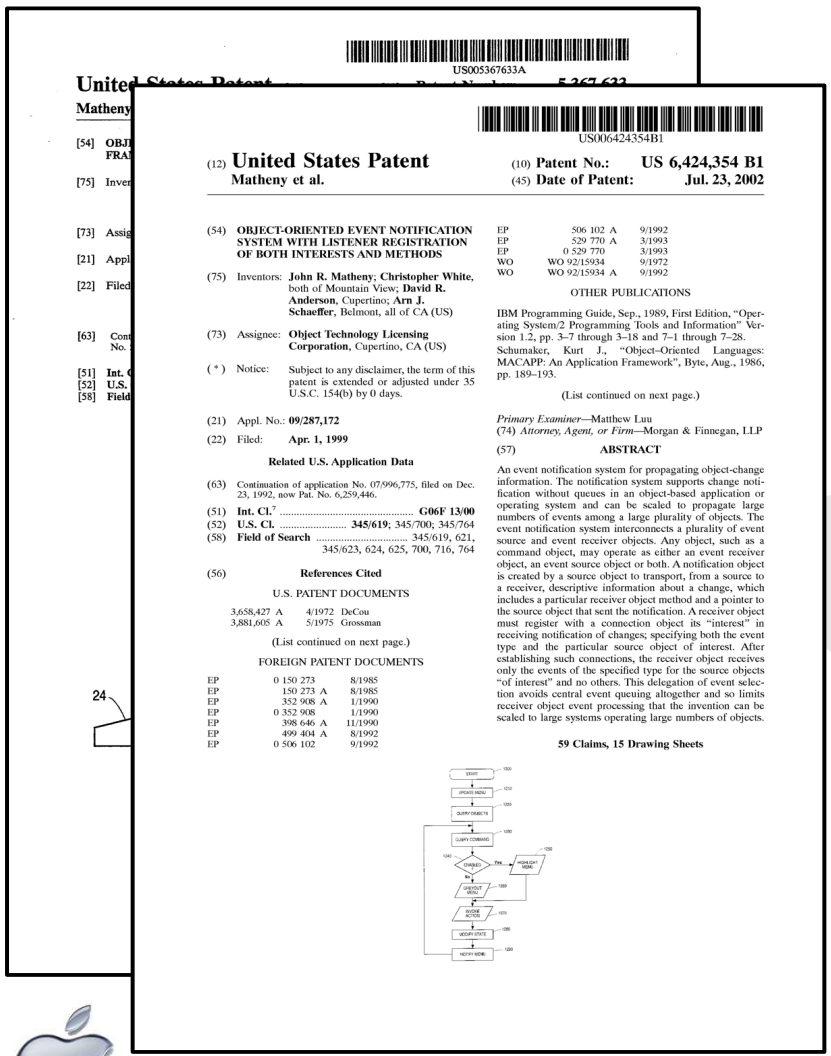
U.S. 5,911,067 (Application Control Transfer)

U.S. 6,067,577 (Dynamic Method Resolution)

U.S. 5,519,867 & 5,379,432 (Threads Wrapper)

U.S. 6,275,983 & 6,684,261 (Wrappers Loader)

U.S. 6,424,354 & 5,367,633 (Interest-Based Notification)



US005367633A

United States Patent

Matheny

[54] OBJ... FRA...

[75] Inver...

[73] Assig...

[21] Appl...

[22] Filed...

[63] Cont... No. ...

[51] Int. C...
[52] U.S. ...
[58] Field...

24

US006424354B1

(12) **United States Patent**
Matheny et al.

(10) **Patent No.: US 6,424,354 B1**
(45) **Date of Patent: Jul. 23, 2002**

(54) **OBJECT-ORIENTED EVENT NOTIFICATION SYSTEM WITH LISTENER REGISTRATION OF BOTH INTERESTS AND METHODS**

(75) Inventors: **John R. Matheny**; **Christopher White**, both of Mountain View; **David R. Anderson**, Cupertino; **Arn J. Schaeffer**, Belmont, all of CA (US)

(73) Assignee: **Object Technology Licensing Corporation**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/287,172**

(22) Filed: **Apr. 1, 1999**

Related U.S. Application Data

(63) Continuation of application No. 07/996,775, filed on Dec. 23, 1992, now Pat. No. 6,259,446.

(51) **Int. Cl.**[7] **G06F 13/00**
(52) **U.S. Cl.** **345/619**; 345/700; 345/764
(58) **Field of Search** 345/619, 621, 345/623, 624, 625, 700, 716, 764

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,658,427	A	4/1972	DeCou
3,881,605	A	5/1975	Grossman

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

EP	0 150 273		8/1985
EP	150 273	A	8/1985
EP	352 908	A	1/1990
EP	0 352 908		1/1990
EP	398 646	A	11/1990
EP	499 404	A	8/1992
EP	0 506 102		9/1992
EP	506 102	A	9/1992
EP	529 770	A	3/1993
EP	0 529 770		3/1993
WO	WO 92/15934		9/1972
WO	WO 92/15934	A	9/1992

OTHER PUBLICATIONS

IBM Programming Guide, Sep., 1989, First Edition, "Operating System/2 Programming Tools and Information" Version 1.2, pp. 3–7 through 3–18 and 7–1 through 7–28.
Schumaker, Kurt J., "Object–Oriented Languages: MACAPP: An Application Framework", Byte, Aug., 1986, pp. 189–193.

(List continued on next page.)

Primary Examiner—Matthew Luu
(74) *Attorney, Agent, or Firm*—Morgan & Finnegan, LLP

(57) **ABSTRACT**

An event notification system for propagating object-change information. The notification system supports change notification without queues in an object-based application or operating system and can be scaled to propagate large numbers of events among a large plurality of objects. The event notification system interconnects a plurality of event source and event receiver objects. Any object, such as a command object, may operate as either an event receiver object, an event source object or both. A notification object is created by a source object to transport, from a source to a receiver, descriptive information about a change, which includes a particular receiver object method and a pointer to the source object that sent the notification. A receiver object must register with a connection object its "interest" in receiving notification of changes; specifying both the event type and the particular source object of interest. After establishing such connections, the receiver object receives only the events of the specified type for the source objects "of interest" and no others. This delegation of event selection avoids central event queuing altogether and so limits receiver object event processing that the invention can be scaled to large systems operating large numbers of objects.

59 Claims, 15 Drawing Sheets

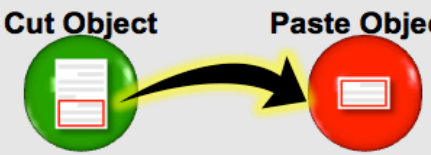
- "Object-Oriented Event Notification System With Listener Registration Of Both Interests And Methods"
- Filing Date: April 1, 1999
  - Priority: December 23, 1992

(57) **ABSTRACT**

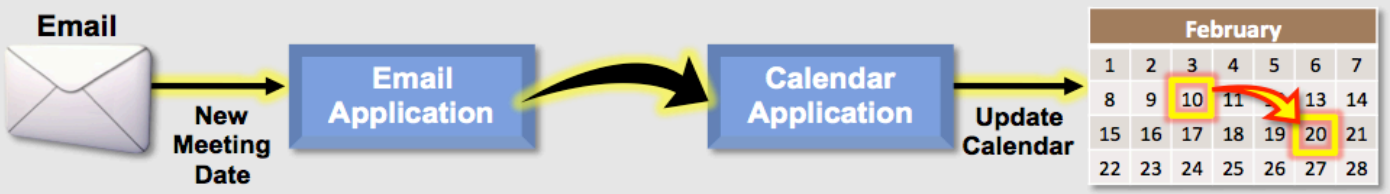
An event notification system for propagating object-change information. The notification system supports change notification without queues in an object-based application or operating system and can be scaled to propagate large numbers of events among a large plurality of objects. The event notification system interconnects a plurality of event source and event receiver objects. Any object, such as a command object, may operate as either an event receiver object, an event source object or both. A notification object is created by a source object to transport, from a source to a receiver, descriptive information about a change, which includes a particular receiver object method and a pointer to the source object that sent the notification. A receiver object must register with a connection object its "interest" in receiving notification of changes; specifying both the event type and the particular source object of interest. After establishing such connections, the receiver object receives only the events of the specified type for the source objects "of interest" and no others. This delegation of event selection avoids central event queuing altogether and so limits receiver object event processing that the invention can be scaled to large systems operating large numbers of objects.

U.S. 6,424,354 & 5,367,633 (Interest-Based Notification)

- **Objects generate new "events" triggering a response**
  - 101001101 001100100 010111010 – **New Data**
  - **Battery level**
  - **Signal strength**
  - **Message arrival**
- **Software objects communicate with each other**



- **Interactive applications communicate with each other**



U.S. 6,424,354 & 5,367,633 (Interest-Based Notification)

Android Code Infringing the '354 and '633 Patents

Event:

instance of android.content.Intent

Receiver object:

instance of android.content.BroadcastReceiver

Connection object:

instance of android.content.Context

Registering for interest:

calling android.content.Context.registerReceiver(...) method

Broadcasting event:

calling android.content.Context.sendBroadcast(...) method

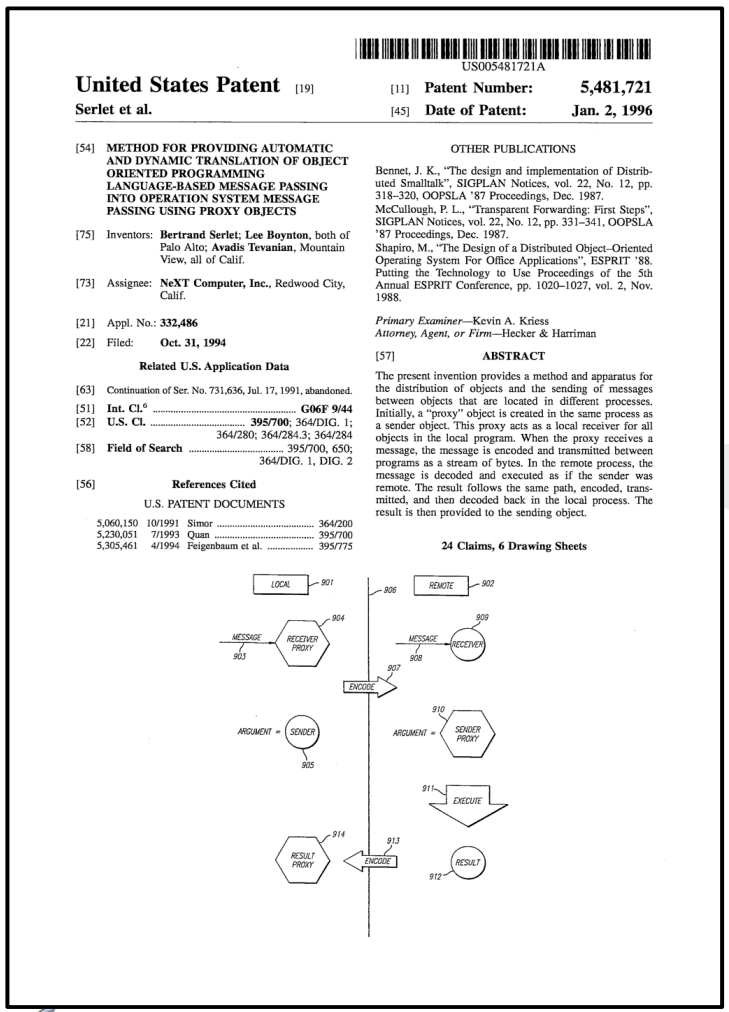
Sender object:

any object calling android.content.Context.sendBroadcast(...)

Method called upon receive:

android.content.BroadcastReceiver.onReceive(...) method

U.S. 5,481,721 (Binder)



US005481721A

United States Patent [19]
Serlet et al.

[11] **Patent Number: 5,481,721**
[45] **Date of Patent: Jan. 2, 1996**

[54] **METHOD FOR PROVIDING AUTOMATIC AND DYNAMIC TRANSLATION OF OBJECT ORIENTED PROGRAMMING LANGUAGE-BASED MESSAGE PASSING INTO OPERATION SYSTEM MESSAGE PASSING USING PROXY OBJECTS**

[75] Inventors: **Bertrand Serlet**; **Lee Boynton**, both of Palo Alto; **Avadis Tevanian**, Mountain View, all of Calif.

[73] Assignee: **NeXT Computer, Inc.**, Redwood City, Calif.

[21] Appl. No.: **332,486**

[22] Filed: **Oct. 31, 1994**

Related U.S. Application Data

[63] Continuation of Ser. No. 731,636, Jul. 17, 1991, abandoned.

[51] **Int. Cl.**[6] **G06F 9/44**

[52] **U.S. Cl.** **395/700**; 364/DIG. 1; 364/280; 364/284.3; 364/284

[58] **Field of Search** ..... 395/700, 650; 364/DIG. 1, DIG. 2

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,060,150	10/1991	Simor	364/200
5,230,051	7/1993	Quan	395/700
5,305,461	4/1994	Feigenbaum et al.	395/775

OTHER PUBLICATIONS

Bennet, J. K., "The design and implementation of Distributed Smalltalk", SIGPLAN Notices, vol. 22, No. 12, pp. 318–320, OOPSLA '87 Proceedings, Dec. 1987.

McCullough, P. L., "Transparent Forwarding: First Steps", SIGPLAN Notices, vol. 22, No. 12, pp. 331–341, OOPSLA '87 Proceedings, Dec. 1987.

Shapiro, M., "The Design of a Distributed Object–Oriented Operating System For Office Applications", ESPRIT '88. Putting the Technology to Use Proceedings of the 5th Annual ESPRIT Conference, pp. 1020–1027, vol. 2, Nov. 1988.

Primary Examiner—Kevin A. Kriess
Attorney, Agent, or Firm—Hecker & Harriman

[57] **ABSTRACT**

The present invention provides a method and apparatus for the distribution of objects and the sending of messages between objects that are located in different processes. Initially, a "proxy" object is created in the same process as a sender object. This proxy acts as a local receiver for all objects in the local program. When the proxy receives a message, the message is encoded and transmitted between programs as a stream of bytes. In the remote process, the message is decoded and executed as if the sender was remote. The result follows the same path, encoded, transmitted, and then decoded back in the local process. The result is then provided to the sending object.

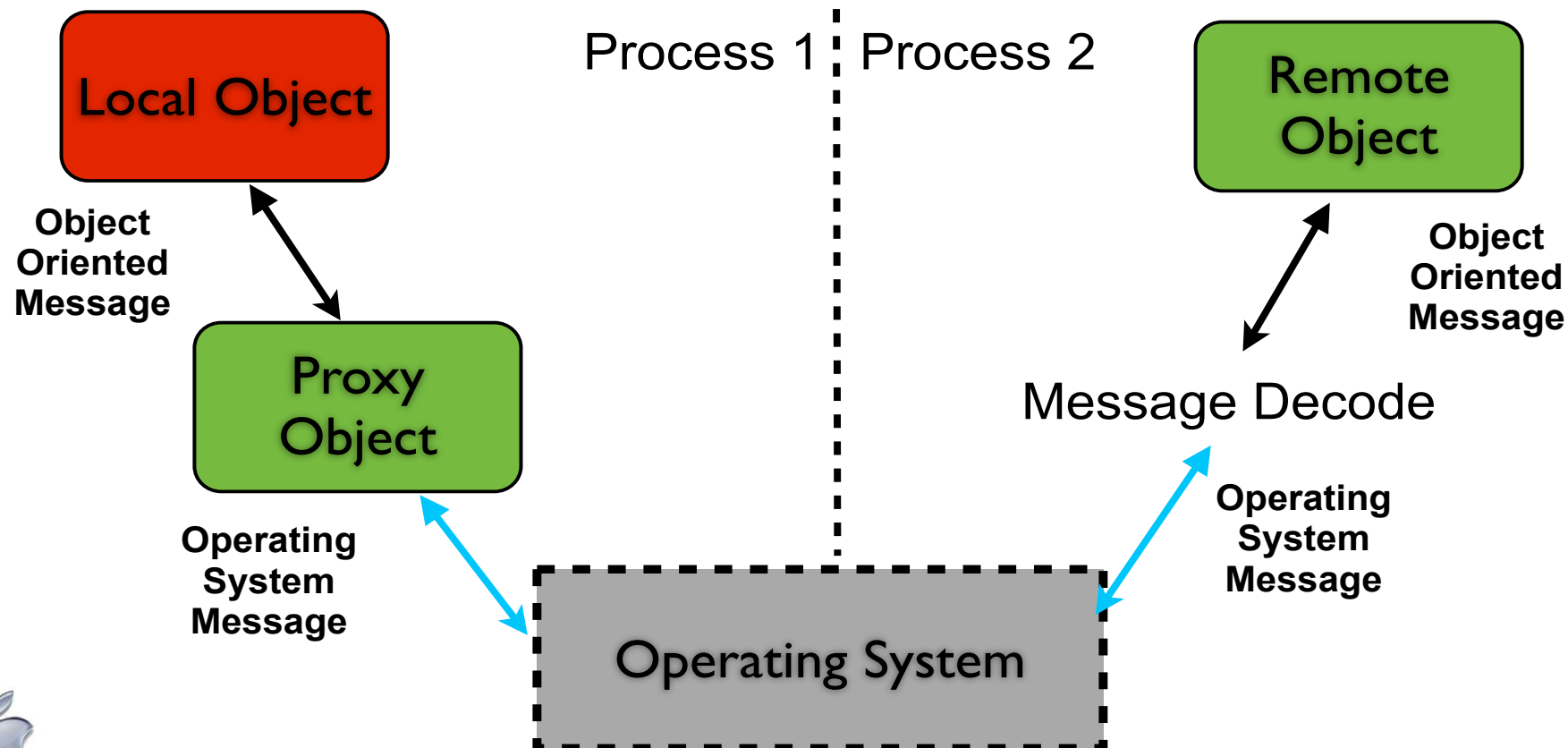
24 Claims, 6 Drawing Sheets

- "Method For Providing Automatic And Dynamic Translation Of Object Oriented Programming Language-Based Message Passing Into Operation System Message Passing Using Proxy Objects"
- Filing Date: October 31, 1994
  - Priority: July 17, 1991

> [57] **ABSTRACT**
>
> The present invention provides a method and apparatus for the distribution of objects and the sending of messages between objects that are located in different processes. Initially, a "proxy" object is created in the same process as a sender object. This proxy acts as a local receiver for all objects in the local program. When the proxy receives a message, the message is encoded and transmitted between programs as a stream of bytes. In the remote process, the message is decoded and executed as if the sender was remote. The result follows the same path, encoded, transmitted, and then decoded back in the local process. The result is then provided to the sending object.

U.S. 5,481,721 (Binder)

- The '721 patent covers an object-oriented IPC mechanism
  - The local object sends messages to the proxy, and the proxy converts them to OS messages so they can be sent to the remote object
- Android's Binder IPC uses the '721 patent
  - Android uses IPC extensively to let applications access system services



U.S. 5,481,721 (Binder)

Android's Binder IPC Infringes the '721 Patent

Android.OS.Binder

public class
Binder

Summary: Inherited Constants | Ctors | Methods | Protected Methods | Inherited Methods | [Expand All]
Since: API Level 1

extends Object
implements IBinder

java.lang.Object
↳android.os.Binder

Class Overview

Base class for a remotable object, the core part of a lightweight remote procedure call mechanism defined by IBinder. This class is an implementation of IBinder that provides the standard support creating a local implementation of such an object.

Most developers will not implement this class directly, instead using the aidl tool to describe the desired interface, having it generate the appropriate Binder subclass. You can, however, derive directly from Binder to implement your own custom RPC protocol or simply instantiate a raw Binder object directly to use as a token that can be shared across processes.

See Also
IBinder

Android.OS.IBinder

public interface
IBinder

Summary: Nested Classes | Constants | Methods | [Expand All]
Since: API Level 1

android.os.IBinder

Known Indirect Subclasses
Binder

Class Overview

Base interface for a remotable object, the core part of a lightweight remote procedure call mechanism designed for high performance when performing in-process and cross-process calls. This interface describes the abstract protocol for interacting with a remotable object. Do not implement this interface directly, instead extend from Binder.

The key IBinder API is transact() matched by Binder.onTransact(). These methods allow you to send a call to an IBinder object and receive a call coming in to a Binder object, respectively. This transaction API is synchronous, such that a call to transact() does not return until the target has returned from Binder.onTransact(); this is the expected behavior when calling an object that exists in the local process, and the underlying inter-process communication (IPC) mechanism ensures that these same semantics apply when going across processes.

The data sent through transact() is a Parcel, a generic buffer of data that also maintains some meta-data about its contents. The meta data is used to manage IBinder object references in the buffer, so that those references can be maintained as the buffer moves across processes. This mechanism ensures that when an IBinder is written into a Parcel and sent to another process, if that other process sends a reference to that same IBinder back to the original process, then the original process will receive the same IBinder object back. These semantics allow IBinder/Binder objects to be used as a unique identity (to serve as a token or for other purposes) that can be managed across processes.

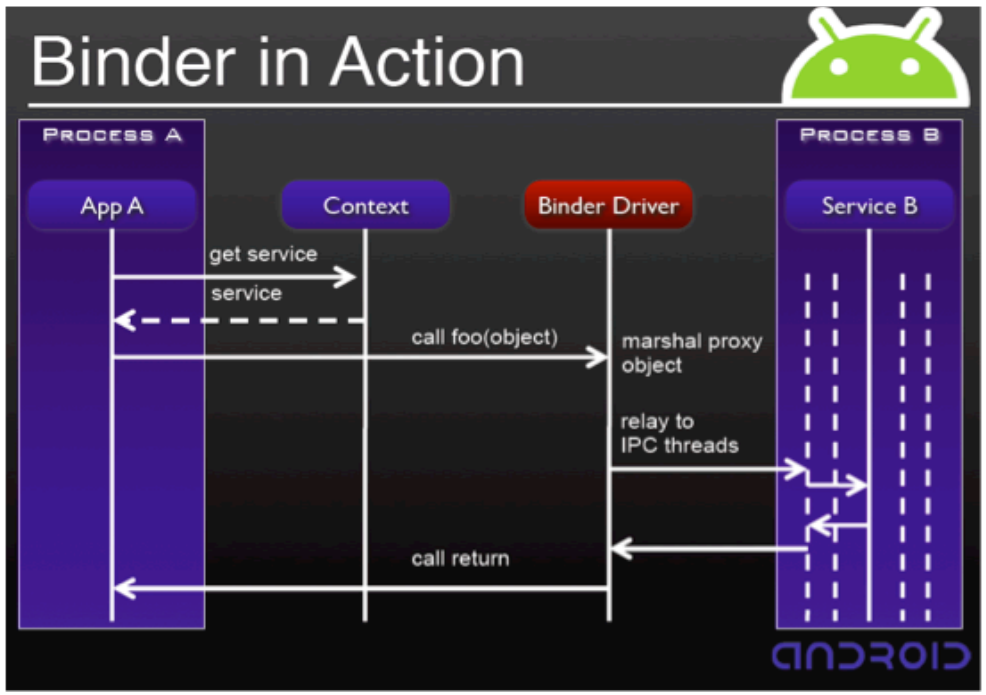
The system maintains a pool of transaction threads in each process that it runs in. These threads are used to dispatch all IPCs coming in from other processes. For example, when an IPC is made from process A to process B, the calling thread in A blocks in transact() as it sends the transaction to process B. The next available pool thread in B receives the incoming transaction, calls Binder.onTransact() on the target object, and replies with the result Parcel. Upon receiving its result, the thread in process A returns to allow its execution to continue. In effect, other processes appear to use as additional threads that you did not create executing in your own process.

The Binder system also supports recursion across processes. For example if process A performs a transaction to process B, and process B while handling that transaction calls transact() on an IBinder that is implemented in A, then the thread in A that is currently waiting for the original transaction to finish will take care of calling Binder.onTransact() on the object being called by B. This ensures that the recursion semantics when calling remote binder object are the same as when calling local objects.

When working with remote objects, you often want to find out when they are no longer valid. There are three ways this can be determined:

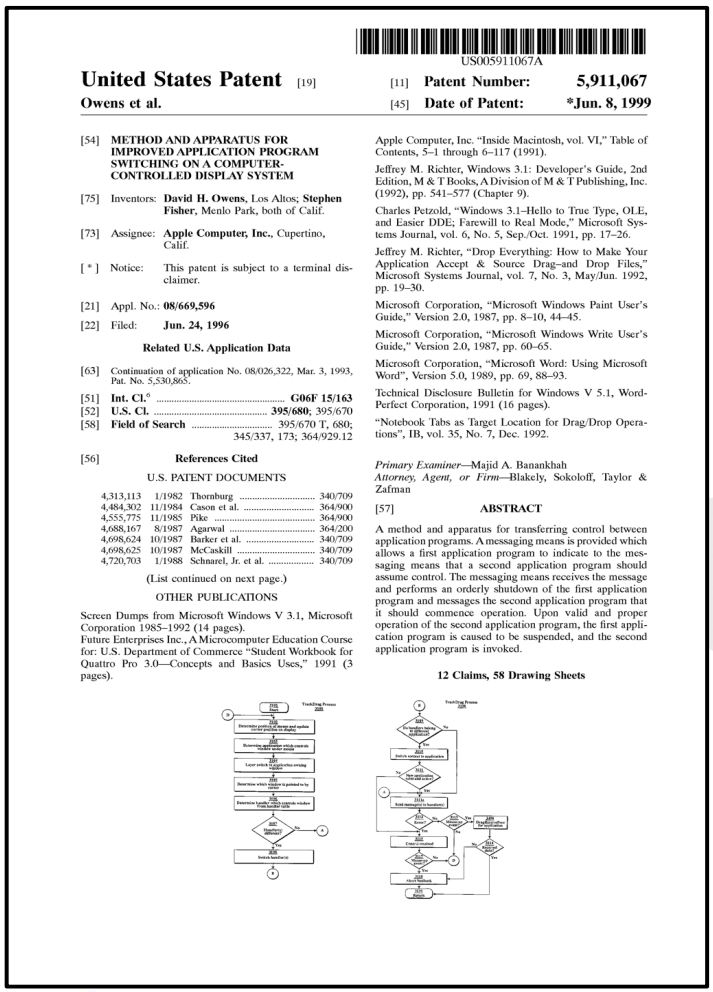
- The transact() method will throw a RemoteException exception if you try to call it on an IBinder whose process no longer exists.
- The pingBinder() method can be called, and will return false if the remote process no longer exists.
- The linkToDeath() method can be used to register a IBinder.DeathRecipient with the IBinder, which will be called when its containing process goes away.

See Also
Binder



U.S. 5,911,067 (Application Control Transfer)

- "Method And Apparatus For Improved Application Program Switching On A Computer-Controlled Display System"
- Filing Date: June 24, 1996
  - Priority: March 3, 1993



US005911067A

United States Patent [19]
Owens et al.

[11] **Patent Number: 5,911,067**
[45] **Date of Patent: *Jun. 8, 1999**

[54] **METHOD AND APPARATUS FOR IMPROVED APPLICATION PROGRAM SWITCHING ON A COMPUTER-CONTROLLED DISPLAY SYSTEM**

[75] Inventors: **David H. Owens**, Los Altos; **Stephen Fisher**, Menlo Park, both of Calif.

[73] Assignee: **Apple Computer, Inc.**, Cupertino, Calif.

[*] Notice: This patent is subject to a terminal disclaimer.

[21] Appl. No.: **08/669,596**

[22] Filed: **Jun. 24, 1996**

Related U.S. Application Data

[63] Continuation of application No. 08/026,322, Mar. 3, 1993, Pat. No. 5,530,865.

[51] **Int. Cl.**[6] **G06F 15/163**
[52] **U.S. Cl.** **395/680**; 395/670
[58] **Field of Search** ..... 395/670 T, 680; 345/337, 173; 364/929.12

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,313,113	1/1982	Thornburg	340/709
4,484,302	11/1984	Cason et al.	364/900
4,555,775	11/1985	Pike	364/900
4,688,167	8/1987	Agarwal	364/200
4,698,624	10/1987	Barker et al.	340/709
4,698,625	10/1987	McCaskill	340/709
4,720,703	1/1988	Schnarel, Jr. et al.	340/709

(List continued on next page.)

OTHER PUBLICATIONS

Screen Dumps from Microsoft Windows V 3.1, Microsoft Corporation 1985–1992 (14 pages).
Future Enterprises Inc., A Microcomputer Education Course for: U.S. Department of Commerce "Student Workbook for Quattro Pro 3.0—Concepts and Basics Uses," 1991 (3 pages).

Apple Computer, Inc. "Inside Macintosh, vol. VI," Table of Contents, 5–1 through 6–117 (1991).

Jeffrey M. Richter, Windows 3.1: Developer's Guide, 2nd Edition, M & T Books, A Division of M & T Publishing, Inc. (1992), pp. 541–577 (Chapter 9).

Charles Petzold, "Windows 3.1–Hello to True Type, OLE, and Easier DDE; Farewill to Real Mode," Microsoft Systems Journal, vol. 6, No. 5, Sep./Oct. 1991, pp. 17–26.

Jeffrey M. Richter, "Drop Everything: How to Make Your Application Accept & Source Drag–and Drop Files," Microsoft Systems Journal, vol. 7, No. 3, May/Jun. 1992, pp. 19–30.

Microsoft Corporation, "Microsoft Windows Paint User's Guide," Version 2.0, 1987, pp. 8–10, 44–45.

Microsoft Corporation, "Microsoft Windows Write User's Guide," Version 2.0, 1987, pp. 60–65.

Microsoft Corporation, "Microsoft Word: Using Microsoft Word", Version 5.0, 1989, pp. 69, 88–93.

Technical Disclosure Bulletin for Windows V 5.1, WordPerfect Corporation, 1991 (16 pages).

"Notebook Tabs as Target Location for Drag/Drop Operations", IB, vol. 35, No. 7, Dec. 1992.

Primary Examiner—Majid A. Banankhah
Attorney, Agent, or Firm—Blakely, Sokoloff, Taylor & Zafman

[57] **ABSTRACT**

A method and apparatus for transferring control between application programs. A messaging means is provided which allows a first application program to indicate to the messaging means that a second application program should assume control. The messaging means receives the message and performs an orderly shutdown of the first application program and messages the second application program that it should commence operation. Upon valid and proper operation of the second application program, the first application program is caused to be suspended, and the second application program is invoked.

12 Claims, 58 Drawing Sheets

[57] **ABSTRACT**

A method and apparatus for transferring control between application programs. A messaging means is provided which allows a first application program to indicate to the messaging means that a second application program should assume control. The messaging means receives the message and performs an orderly shutdown of the first application program and messages the second application program that it should commence operation. Upon valid and proper operation of the second application program, the first application program is caused to be suspended, and the second application program is invoked.

U.S. 5,911,067 (Application Control Transfer)

- '067 patent describes transferring control from a first application program to a second application program
- The Android System Process calls onPause ( ), onCreate ( ), onStart ( ), onResume( ), and onStop( ) to coordinate switching between applications

Coordinating activities

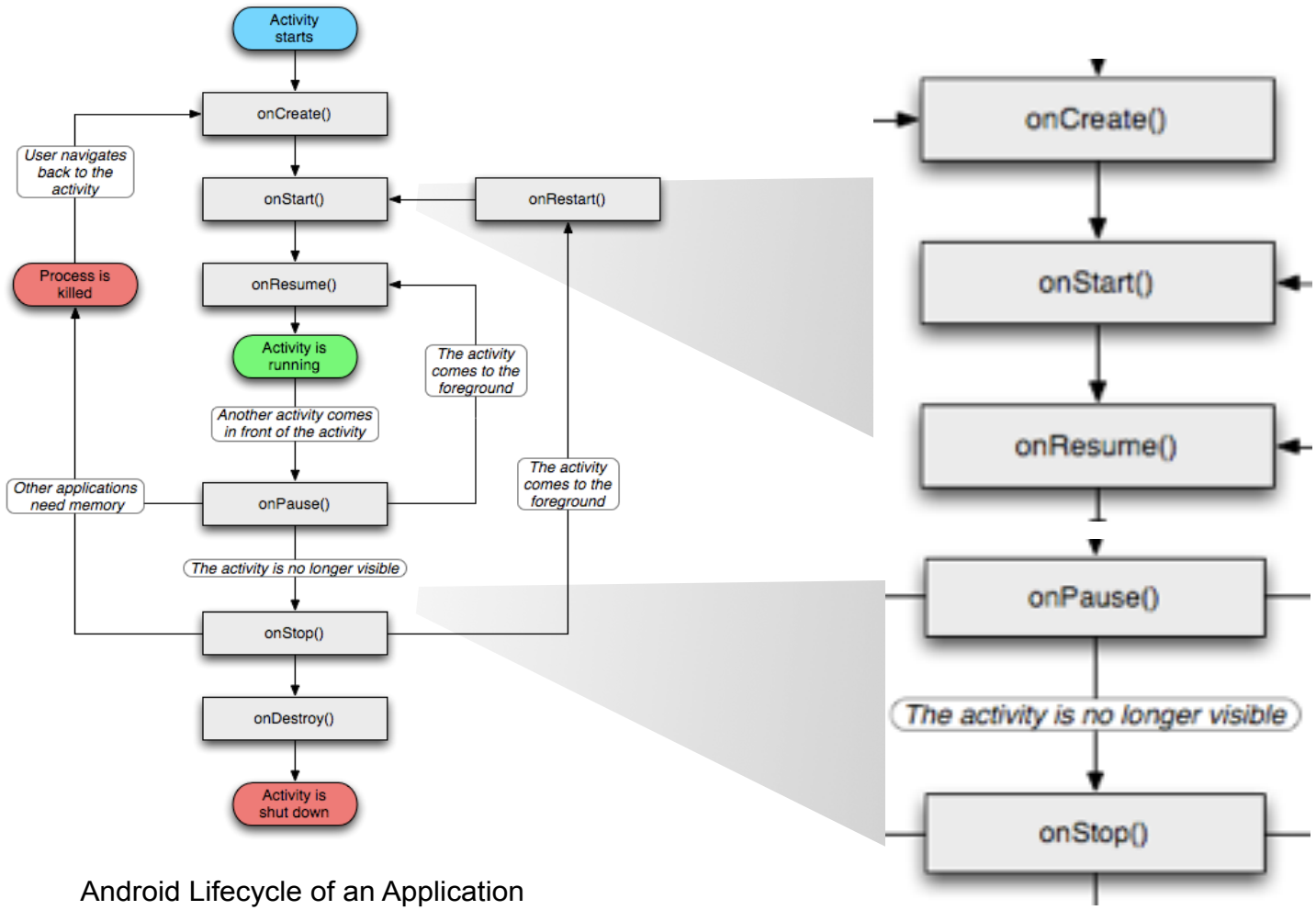
When one activity starts another, they both experience lifecycle transitions. One pauses and may stop, while the other starts up. On occasion, you may need to coordinate these activities, one with the other.

The order of lifecycle callbacks is well defined, particularly when the two activities are in the same process:

1. The current activity's `onPause()` method is called.
2. Next, the starting activity's `onCreate()`, `onStart()`, and `onResume()` methods are called in sequence.
3. Then, if the starting activity is no longer visible on screen, its `onStop()` method is called.

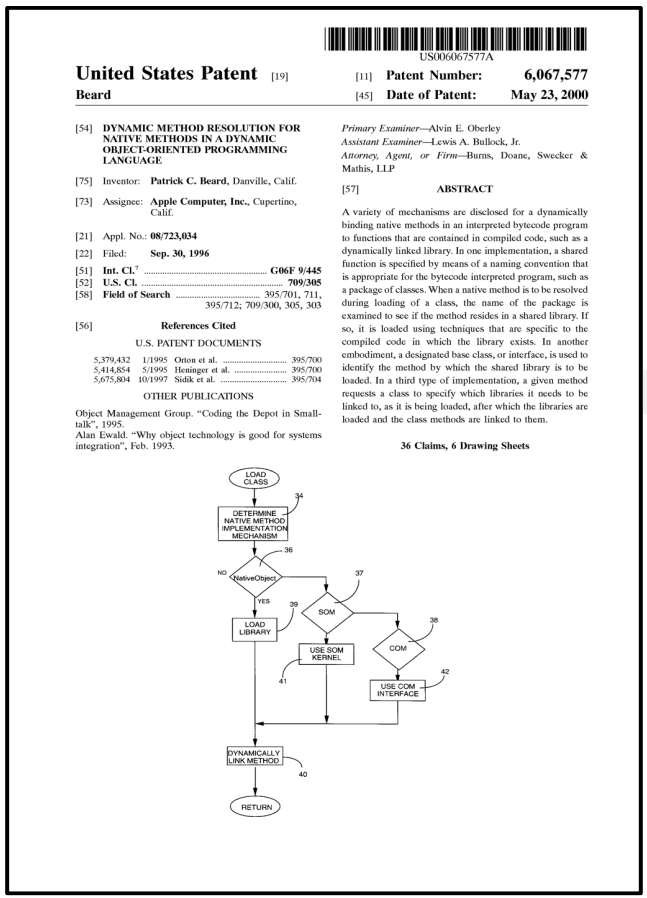
http://developer.android.com/guide/topics/fundamentals.html#lcycles

U.S. 5,911,067 (Application Control Transfer)



Android Lifecycle of an Application

U.S. 6,067,577 (Dynamic Method Resolution)



US006067577A

United States Patent [19]
Beard

[11] **Patent Number: 6,067,577**
[45] **Date of Patent: May 23, 2000**

[54] **DYNAMIC METHOD RESOLUTION FOR NATIVE METHODS IN A DYNAMIC OBJECT-ORIENTED PROGRAMMING LANGUAGE**

[75] Inventor: **Patrick C. Beard**, Danville, Calif.

[73] Assignee: **Apple Computer, Inc.**, Cupertino, Calif.

[21] Appl. No.: **08/723,034**

[22] Filed: **Sep. 30, 1996**

[51] **Int. Cl.**[7] **G06F 9/445**
[52] **U.S. Cl.** **709/305**
[58] **Field of Search** 395/701, 711, 395/712; 709/300, 305, 303

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,379,432	1/1995	Orton et al.	395/700
5,414,854	5/1995	Heninger et al.	395/700
5,675,804	10/1997	Sidik et al.	395/704

OTHER PUBLICATIONS

Object Management Group. "Coding the Depot in Small-talk", 1995.
Alan Ewald. "Why object technology is good for systems integration", Feb. 1993.

Primary Examiner—Alvin E. Oberley
Assistant Examiner—Lewis A. Bullock, Jr.
Attorney, Agent, or Firm—Burns, Doane, Swecker & Mathis, LLP

[57] **ABSTRACT**

A variety of mechanisms are disclosed for a dynamically binding native methods in an interpreted bytecode program to functions that are contained in compiled code, such as a dynamically linked library. In one implementation, a shared function is specified by means of a naming convention that is appropriate for the bytecode interpreted program, such as a package of classes. When a native method is to be resolved during loading of a class, the name of the package is examined to see if the method resides in a shared library. If so, it is loaded using techniques that are specific to the compiled code in which the library exists. In another embodiment, a designated base class, or interface, is used to identify the method by which the shared library is to be loaded. In a third type of implementation, a given method requests a class to specify which libraries it needs to be linked to, as it is being loaded, after which the libraries are loaded and the class methods are linked to them.

36 Claims, 6 Drawing Sheets

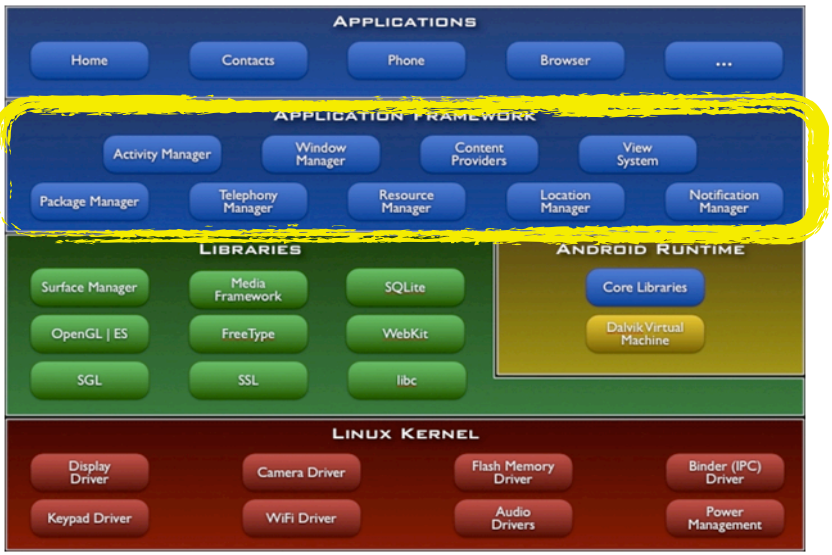
- "Dynamic Method Resolution For Native Methods In A Dynamic Object-Oriented Programming Language"
- Filing Date: September 30, 1996

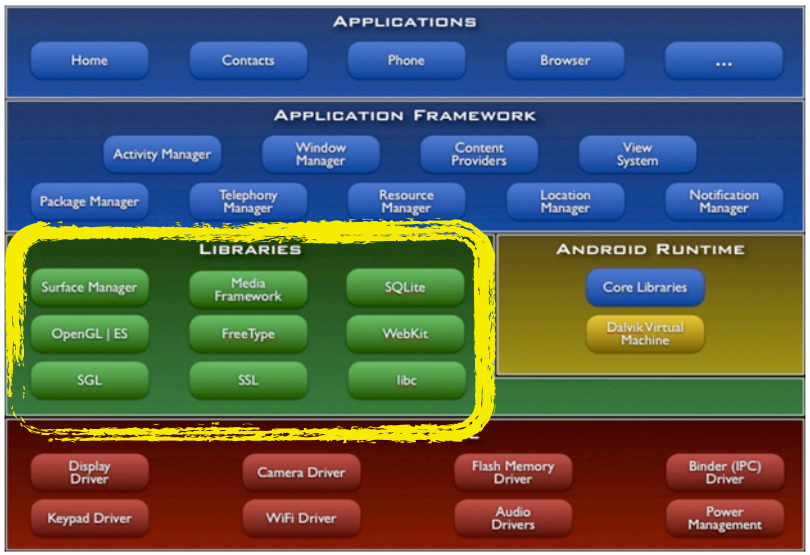
[57] **ABSTRACT**

A variety of mechanisms are disclosed for a dynamically binding native methods in an interpreted bytecode program to functions that are contained in compiled code, such as a dynamically linked library. In one implementation, a shared function is specified by means of a naming convention that is appropriate for the bytecode interpreted program, such as a package of classes. When a native method is to be resolved during loading of a class, the name of the package is examined to see if the method resides in a shared library. If so, it is loaded using techniques that are specific to the compiled code in which the library exists. In another embodiment, a designated base class, or interface, is used to identify the method by which the shared library is to be loaded. In a third type of implementation, a given method requests a class to specify which libraries it needs to be linked to, as it is being loaded, after which the libraries are loaded and the class methods are linked to them.

U.S. 6,067,577 (Dynamic Method Resolution)

'577 patent describes dynamically binding native methods in a first interpreted bytecode language to functions contained in a shared library written in a second program language

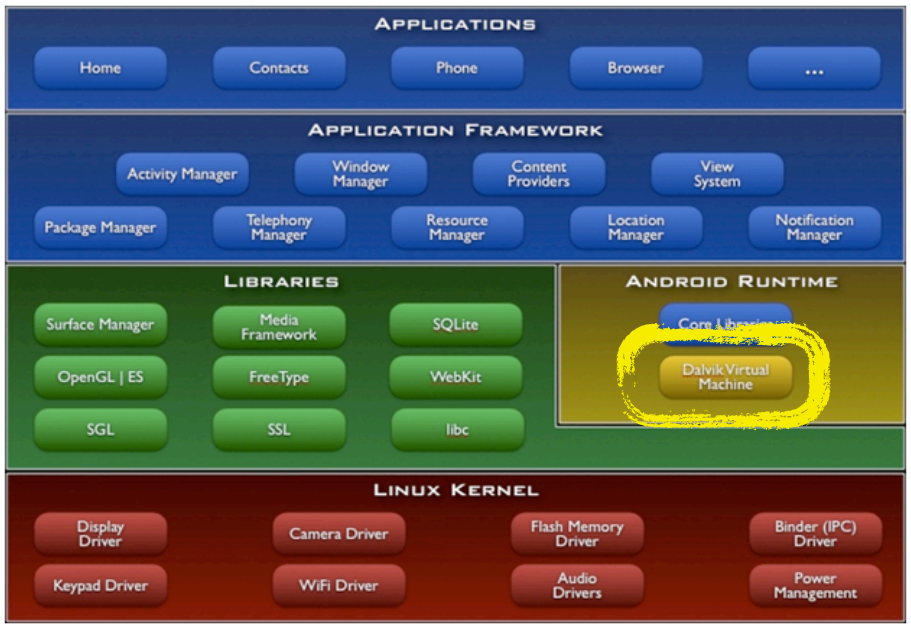




"First interpreted bytecode program language" is Java. Blue boxes in the Android architecture diagram are java.

"Second program language" is C/C++. Green boxes in the Android architecture diagram are C/C++.

U.S. 6,067,577 (Dynamic Method Resolution)



Android defines its own bytecodes called “Dalvik bytecodes.” Dalvik Virtual Machine is a bytecode interpreter.

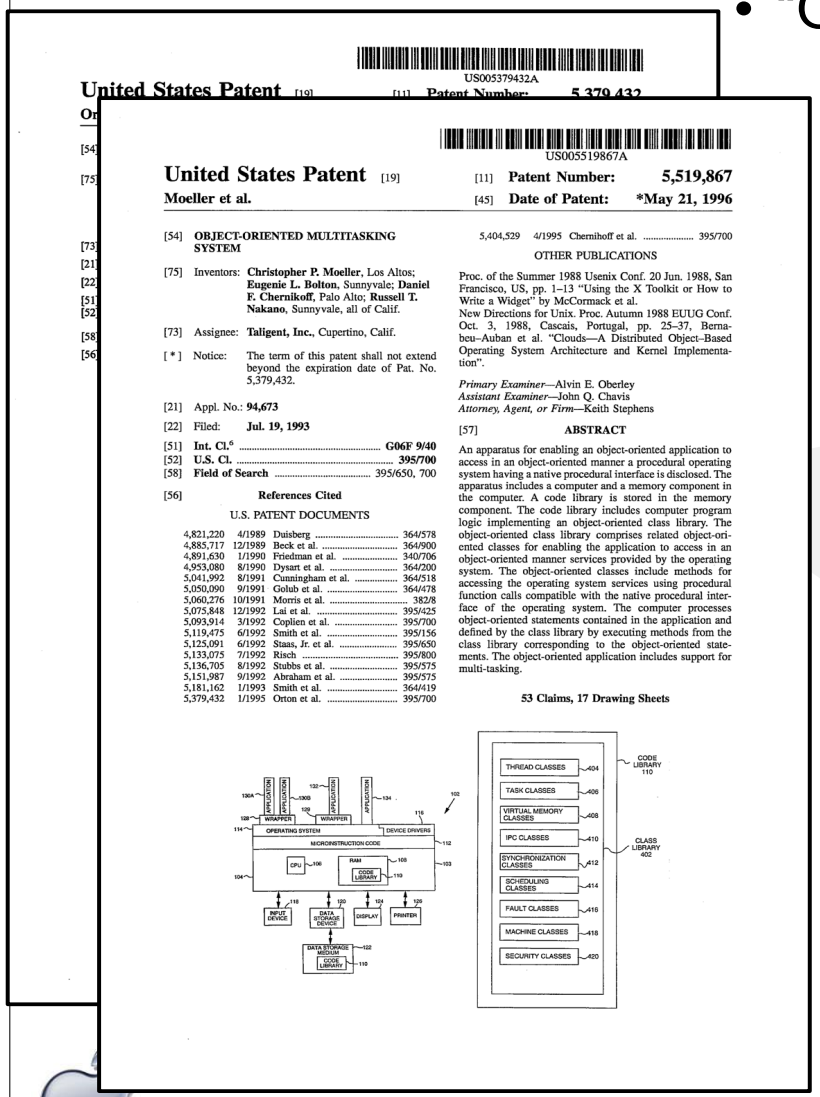
The mechanism in the first program language which “associates a reference to a class of objects and the shared library” is the Java Native Interface ("JNI")

“You can load native code from shared libraries with the standard System.loadLibrary() call.”

http://www.netmite.com/android/mydroid/dalvik/docs/jni-tips.html

U.S. 5,519,867 & 5,379,432 (Threads Wrapper)

- "Object-Oriented Multitasking System"
  - Filing Date: July 19, 1993



US005379432A

United States Patent [19]

[11] Patent Number: 5,379,432

US005519867A

United States Patent [19]

Moeller et al.

[11] **Patent Number: 5,519,867**

[45] **Date of Patent: *May 21, 1996**

[54] **OBJECT-ORIENTED MULTITASKING SYSTEM**

[75] Inventors: **Christopher P. Moeller**, Los Altos; **Eugenie L. Bolton**, Sunnyvale; **Daniel F. Chernikoff**, Palo Alto; **Russell T. Nakano**, Sunnyvale, all of Calif.

[73] Assignee: **Taligent, Inc.**, Cupertino, Calif.

[*] Notice: The term of this patent shall not extend beyond the expiration date of Pat. No. 5,379,432.

[21] Appl. No.: **94,673**

[22] Filed: **Jul. 19, 1993**

[51] **Int. Cl.**[6] **G06F 9/40**

[52] **U.S. Cl.** **395/700**

[58] **Field of Search** 395/650, 700

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,821,220	4/1989	Duisberg	364/578
4,885,717	12/1989	Beck et al.	364/900
4,891,630	1/1990	Friedman et al.	340/706
4,953,080	8/1990	Dysart et al.	364/200
5,041,992	8/1991	Cunningham et al.	364/518
5,050,090	9/1991	Golub et al.	364/478
5,060,276	10/1991	Morris et al.	382/8
5,075,848	12/1992	Lai et al.	395/425
5,093,914	3/1992	Coplien et al.	395/700
5,119,475	6/1992	Smith et al.	395/156
5,125,091	6/1992	Staas, Jr. et al.	395/650
5,133,075	7/1992	Risch	395/800
5,136,705	8/1992	Stubbs et al.	395/575
5,151,987	9/1992	Abraham et al.	395/575
5,181,162	1/1993	Smith et al.	364/419
5,379,432	1/1995	Orton et al.	395/700
5,404,529	4/1995	Chernihoff et al.	395/700

OTHER PUBLICATIONS

Proc. of the Summer 1988 Usenix Conf. 20 Jun. 1988, San Francisco, US, pp. 1–13 "Using the X Toolkit or How to Write a Widget" by McCormack et al.

New Directions for Unix. Proc. Autumn 1988 EUUG Conf. Oct. 3, 1988, Cascais, Portugal, pp. 25–37, Bernabeu–Auban et al. "Clouds—A Distributed Object–Based Operating System Architecture and Kernel Implementation".

Primary Examiner—Alvin E. Oberley
Assistant Examiner—John Q. Chavis
Attorney, Agent, or Firm—Keith Stephens

[57] **ABSTRACT**

An apparatus for enabling an object-oriented application to access in an object-oriented manner a procedural operating system having a native procedural interface is disclosed. The apparatus includes a computer and a memory component in the computer. A code library is stored in the memory component. The code library includes computer program logic implementing an object-oriented class library. The object-oriented class library comprises related object-oriented classes for enabling the application to access in an object-oriented manner services provided by the operating system. The object-oriented classes include methods for accessing the operating system services using procedural function calls compatible with the native procedural interface of the operating system. The computer processes object-oriented statements contained in the application and defined by the class library by executing methods from the class library corresponding to the object-oriented statements. The object-oriented application includes support for multi-tasking.

53 Claims, 17 Drawing Sheets

[57] **ABSTRACT**

An apparatus for enabling an object-oriented application to access in an object-oriented manner a procedural operating system having a native procedural interface is disclosed. The apparatus includes a computer and a memory component in the computer. A code library is stored in the memory component. The code library includes computer program logic implementing an object-oriented class library. The object-oriented class library comprises related object-oriented classes for enabling the application to access in an object-oriented manner services provided by the operating system. The object-oriented classes include methods for accessing the operating system services using procedural function calls compatible with the native procedural interface of the operating system. The computer processes object-oriented statements contained in the application and defined by the class library by executing methods from the class library corresponding to the object-oriented statements. The object-oriented application includes support for multi-tasking.

U.S. 5,519,867 & 5,379,432 (Threads Wrapper)

Relates to execution of object-oriented programs on a computer with a procedural operating system

Solves the problem of allowing an object-oriented application to access services of a procedural operating system

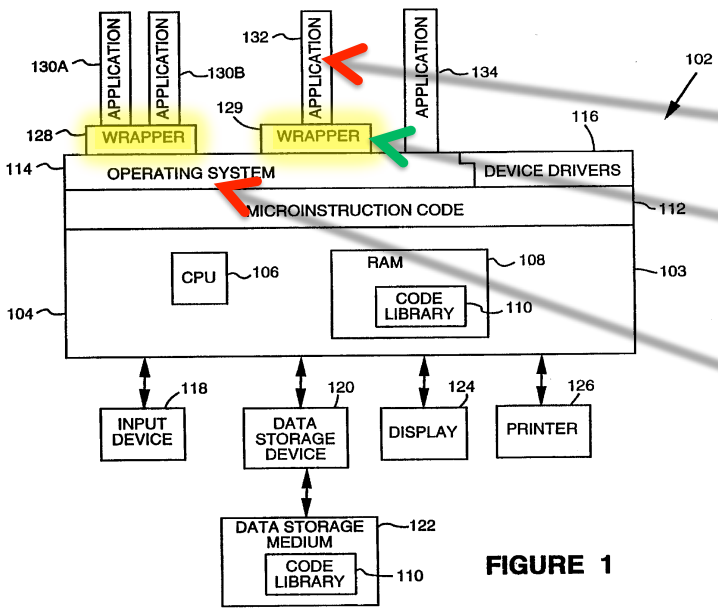


FIGURE 1

Object Oriented Application

Wrapper

Procedural Operating System

U.S. 5,519,867 & 5,379,432 (Threads Wrapper)

Android's Thread and ThreadGroup Classes Infringe the '867 and '432 Patents

java.lang.ThreadGroup

public class
ThreadGroup
Summary: Ctors | Methods | Inherited Methods | [Expand All]
Since: API Level 1

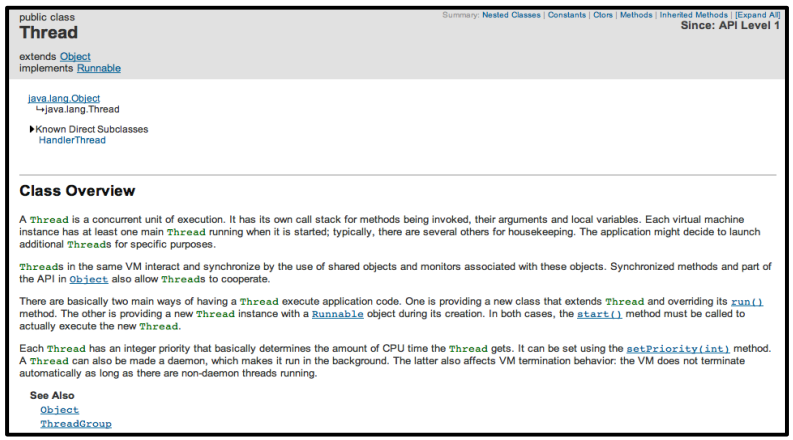
extends Object
implements Thread.UncaughtExceptionHandler

java.lang.Object
↳java.lang.ThreadGroup

Class Overview

A `ThreadGroup` is a means of organizing `Threads` into a hierarchical structure. A `ThreadGroup` can contain zero or more `Threads` and zero or more other `ThreadGroups`. Each `Thread` and each `ThreadGroup` (except the root group) has a unique parent `ThreadGroup`. The result is a tree whose inner nodes are `ThreadGroups` and whose leaf nodes are `Threads`. The unique root of the tree is a `ThreadGroup` that is created at VM startup and has the name "system". The benefit of using `ThreadGroups`, in addition to the mere housekeeping aspect, is that all `Threads` in a `ThreadGroup` can be manipulated together, that is, the `ThreadGroup` has methods that delegate to all its all `Threads`.

See Also
`Thread`
`SecurityManager`



public class
Thread
Summary: Nested Classes | Constants | Ctors | Methods | Inherited Methods | [Expand All]
Since: API Level 1

extends Object
implements Runnable

java.lang.Object
↳java.lang.Thread

Known Direct Subclasses
HandlerThread

Class Overview

A `Thread` is a concurrent unit of execution. It has its own call stack for methods being invoked, their arguments and local variables. Each virtual machine instance has at least one main `Thread` running when it is started; typically, there are several others for housekeeping. The application might decide to launch additional `Threads` for specific purposes.

`Threads` in the same VM interact and synchronize by the use of shared objects and monitors associated with these objects. Synchronized methods and part of the API in `Object` also allow `Threads` to cooperate.

There are basically two main ways of having a `Thread` execute application code. One is providing a new class that extends `Thread` and overriding its `run()` method. The other is providing a new `Thread` instance with a `Runnable` object during its creation. In both cases, the `start()` method must be called to actually execute the new `Thread`.

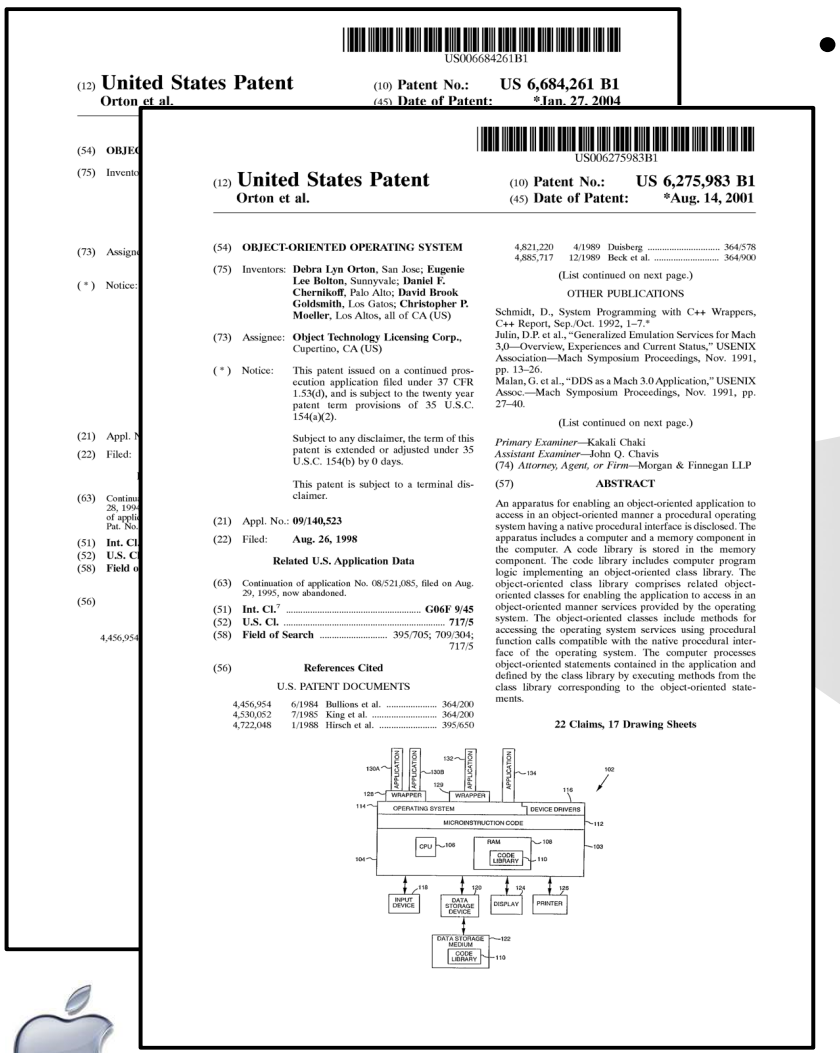
Each `Thread` has an integer priority that basically determines the amount of CPU time the `Thread` gets. It can be set using the `setPriority(int)` method. A `Thread` can also be made a daemon, which makes it run in the background. The latter also affects VM termination behavior: the VM does not terminate automatically as long as there are non-daemon threads running.

See Also
`Object`
`ThreadGroup`

java.lang.Thread

U.S. 6,275,983 & 6,684,261 (Wrappers Loader)

- "Object-Oriented Operating System"
- Filing Date: August 26, 1998
  - Priority: August 29, 1995



US006684261B1

(12) **United States Patent**
Orton et al.

(10) **Patent No.: US 6,684,261 B1**
(45) **Date of Patent: *Jan. 27, 2004**

(54) OBJEC
(75) Invento
(73) Assign
(*) Notice:
(21) Appl. N
(22) Filed:
(63) Continu 28, 199 of appli Pat. No.
(51) Int. Cl
(52) U.S. Cl
(58) Field o
(56)
4,456,954

US006275983B1

(12) **United States Patent**
Orton et al.

(10) **Patent No.: US 6,275,983 B1**
(45) **Date of Patent: *Aug. 14, 2001**

(54) **OBJECT-ORIENTED OPERATING SYSTEM**

(75) Inventors: **Debra Lyn Orton**, San Jose; **Eugenie Lee Bolton**, Sunnyvale; **Daniel F. Chernikoff**, Palo Alto; **David Brook Goldsmith**, Los Gatos; **Christopher P. Moeller**, Los Altos, all of CA (US)

(73) Assignee: **Object Technology Licensing Corp.**, Cupertino, CA (US)

(*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **09/140,523**

(22) Filed: **Aug. 26, 1998**

Related U.S. Application Data

(63) Continuation of application No. 08/521,085, filed on Aug. 29, 1995, now abandoned.

(51) **Int. Cl.**[7] **G06F 9/45**
(52) **U.S. Cl.** **717/5**
(58) **Field of Search** 395/705; 709/304; 717/5

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,456,954	6/1984	Bullions et al.	364/200
4,530,052	7/1985	King et al.	364/200
4,722,048	1/1988	Hirsch et al.	395/650
4,821,220	4/1989	Duisberg	364/578
4,885,717	12/1989	Beck et al.	364/900

(List continued on next page.)

OTHER PUBLICATIONS

Schmidt, D., System Programming with C++ Wrappers, C++ Report, Sep./Oct. 1992, 1–7.*
Julin, D.P. et al., "Generalized Emulation Services for Mach 3,0—Overview, Experiences and Current Status," USENIX Association—Mach Symposium Proceedings, Nov. 1991, pp. 13–26.
Malan, G. et al., "DDS as a Mach 3.0 Application," USENIX Assoc.—Mach Symposium Proceedings, Nov. 1991, pp. 27–40.

(List continued on next page.)

Primary Examiner—Kakali Chaki
Assistant Examiner—John Q. Chavis
(74) *Attorney, Agent, or Firm*—Morgan & Finnegan LLP

(57) **ABSTRACT**

An apparatus for enabling an object-oriented application to access in an object-oriented manner a procedural operating system having a native procedural interface is disclosed. The apparatus includes a computer and a memory component in the computer. A code library is stored in the memory component. The code library includes computer program logic implementing an object-oriented class library. The object-oriented class library comprises related object-oriented classes for enabling the application to access in an object-oriented manner services provided by the operating system. The object-oriented classes include methods for accessing the operating system services using procedural function calls compatible with the native procedural interface of the operating system. The computer processes object-oriented statements contained in the application and defined by the class library by executing methods from the class library corresponding to the object-oriented statements.

22 Claims, 17 Drawing Sheets

(57) **ABSTRACT**

An apparatus for enabling an object-oriented application to access in an object-oriented manner a procedural operating system having a native procedural interface is disclosed. The apparatus includes a computer and a memory component in the computer. A code library is stored in the memory component. The code library includes computer program logic implementing an object-oriented class library. The object-oriented class library comprises related object-oriented classes for enabling the application to access in an object-oriented manner services provided by the operating system. The object-oriented classes include methods for accessing the operating system services using procedural function calls compatible with the native procedural interface of the operating system. The computer processes object-oriented statements contained in the application and defined by the class library by executing methods from the class library corresponding to the object-oriented statements.

U.S. 6,275,983 & 6,684,261 (Wrappers Loader)

- The '983 Patent focuses on dynamic (run-time) loading of wrapper code
  - Executable program logic is loaded into memory at run-time rather than compile-time
  - Program first checks whether code has been loaded into memory, and if not sends a request to the library server
  - Library server loads code into appropriate memory space to make it accessible to the program

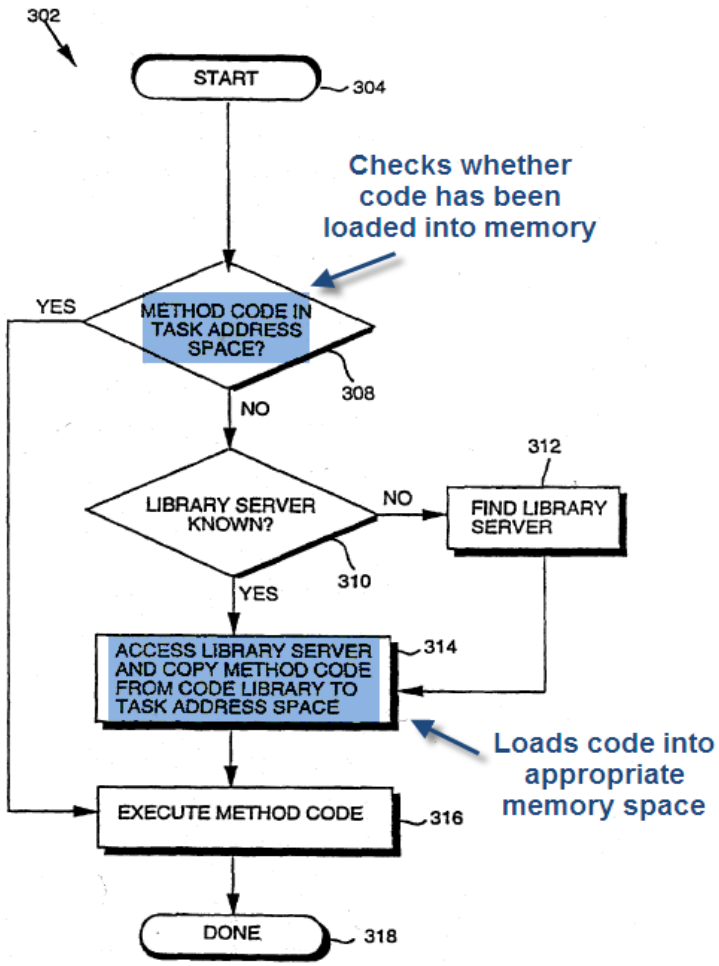
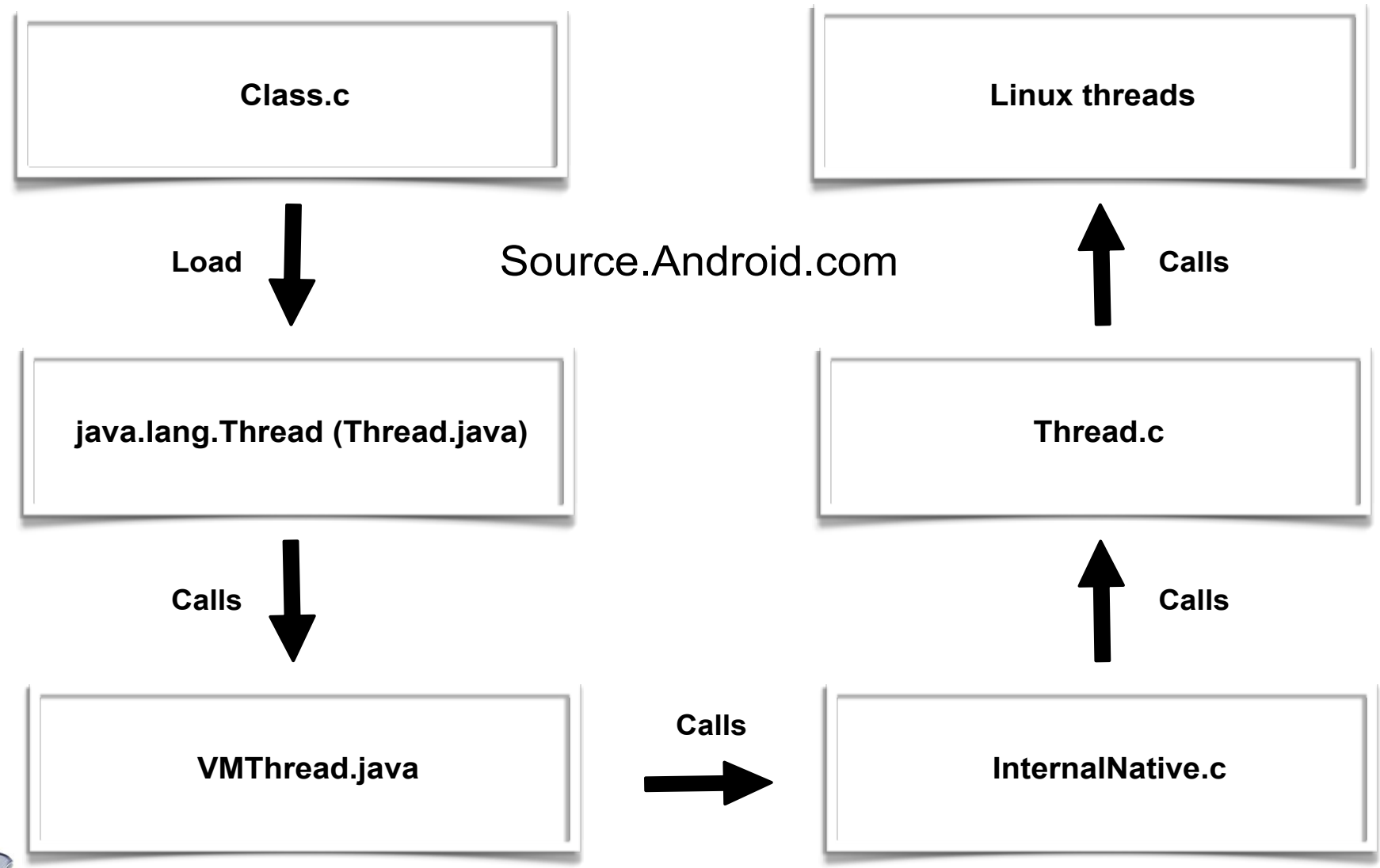
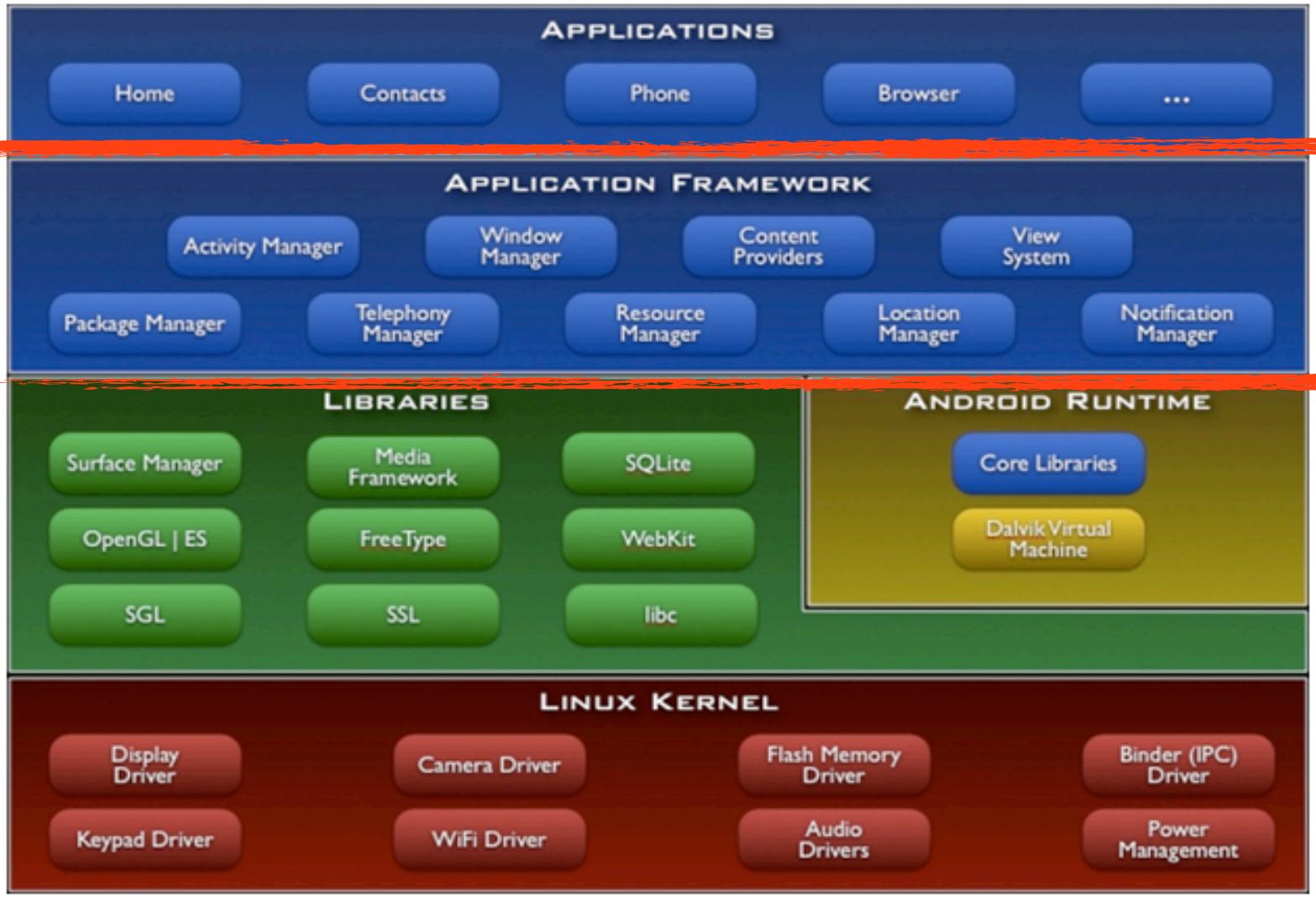


Figure 3

U.S. 6,275,983 & 6,684,261 (Wrappers Loader)



Android Stack



Highlights of Example Apple Patents Infringed by Android Application Framework

U.S. 5,455,599 (Grafport)

U.S. 7,469,381 (Bounce Scrolling)

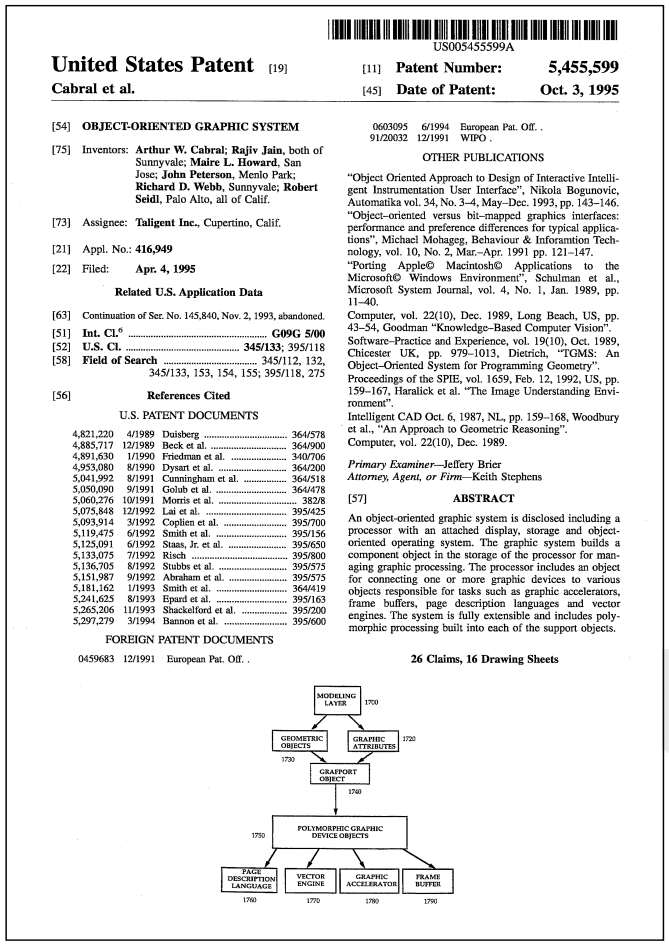
U.S. 5,764,218 (Gesture Detector)

U.S. 6,593,947 (Composite Objects)

U.S. RE41,088 & U.S. 6,956,564 (Rotate Display)

U.S. 7,362,331 (Non-Linear Animation)

U.S. 5,455,599 (Grafport)



US005455599A

United States Patent [19]
Cabral et al.

[11] **Patent Number: 5,455,599**
[45] **Date of Patent: Oct. 3, 1995**

[54] **OBJECT-ORIENTED GRAPHIC SYSTEM**

[75] Inventors: **Arthur W. Cabral**; **Rajiv Jain**, both of Sunnyvale; **Maire L. Howard**, San Jose; **John Peterson**, Menlo Park; **Richard D. Webb**, Sunnyvale; **Robert Seidl**, Palo Alto, all of Calif.

[73] Assignee: **Taligent Inc.**, Cupertino, Calif.

[21] Appl. No.: **416,949**

[22] Filed: **Apr. 4, 1995**

Related U.S. Application Data

[63] Continuation of Ser. No. 145,840, Nov. 2, 1993, abandoned.

[51] **Int. Cl.**[6] **G09G 5/00**

[52] **U.S. Cl.** **345/133**; 395/118

[58] **Field of Search** 345/112, 132, 345/133, 153, 154, 155; 395/118, 275

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,821,220	4/1989	Duisberg	364/578
4,885,717	12/1989	Beck et al.	364/900
4,891,630	1/1990	Friedman et al.	340/706
4,953,080	8/1990	Dysart et al.	364/200
5,041,992	8/1991	Cunningham et al.	364/518
5,050,090	9/1991	Golub et al.	364/478
5,060,276	10/1991	Morris et al.	382/8
5,075,848	12/1992	Lai et al.	395/425
5,093,914	3/1992	Coplien et al.	395/700
5,119,475	6/1992	Smith et al.	395/156
5,125,091	6/1992	Staas, Jr. et al.	395/650
5,133,075	7/1992	Risch	395/800
5,136,705	8/1992	Stubbs et al.	395/575
5,151,987	9/1992	Abraham et al.	395/575
5,181,162	1/1993	Smith et al.	364/419
5,241,625	8/1993	Epard et al.	395/163
5,265,206	11/1993	Shackelford et al.	395/200
5,297,279	3/1994	Bannon et al.	395/600

FOREIGN PATENT DOCUMENTS

0459683	12/1991	European Pat. Off. .
0603095	6/1994	European Pat. Off. .
91/20032	12/1991	WIPO .

OTHER PUBLICATIONS

"Object Oriented Approach to Design of Interactive Intelligent Instrumentation User Interface", Nikola Bogunovic, Automatika vol. 34, No. 3–4, May–Dec. 1993, pp. 143–146.
"Object–oriented versus bit–mapped graphics interfaces: performance and preference differences for typical applications", Michael Mohageg, Behaviour & Inforamtion Technology, vol. 10, No. 2, Mar.–Apr. 1991 pp. 121–147.
"Porting Apple© Macintosh© Applications to the Microsoft© Windows Environment", Schulman et al., Microsoft System Journal, vol. 4, No. 1, Jan. 1989, pp. 11–40.
Computer, vol. 22(10), Dec. 1989, Long Beach, US, pp. 43–54, Goodman "Knowledge–Based Computer Vision".
Software–Practice and Experience, vol. 19(10), Oct. 1989, Chicester UK, pp. 979–1013, Dietrich, "TGMS: An Object–Oriented System for Programming Geometry".
Proceedings of the SPIE, vol. 1659, Feb. 12, 1992, US, pp. 159–167, Haralick et al. "The Image Understanding Environment".
Intelligent CAD Oct. 6, 1987, NL, pp. 159–168, Woodbury et al., "An Approach to Geometric Reasoning".
Computer, vol. 22(10), Dec. 1989.

Primary Examiner—Jeffery Brier
Attorney, Agent, or Firm—Keith Stephens

[57] **ABSTRACT**

An object-oriented graphic system is disclosed including a processor with an attached display, storage and object-oriented operating system. The graphic system builds a component object in the storage of the processor for managing graphic processing. The processor includes an object for connecting one or more graphic devices to various objects responsible for tasks such as graphic accelerators, frame buffers, page description languages and vector engines. The system is fully extensible and includes polymorphic processing built into each of the support objects.

26 Claims, 16 Drawing Sheets

- "Object-Oriented Graphic System"
- Filing Date: April 4, 1995
  - Priority: November 2, 1993

[57] **ABSTRACT**

An object-oriented graphic system is disclosed including a processor with an attached display, storage and object-oriented operating system. The graphic system builds a component object in the storage of the processor for managing graphic processing. The processor includes an object for connecting one or more graphic devices to various objects responsible for tasks such as graphic accelerators, frame buffers, page description languages and vector engines. The system is fully extensible and includes polymorphic processing built into each of the support objects.

U.S. 5,455,599 (Grafport)

15. An apparatus for graphic processing, comprising:

(a) a processor,

(b) a storage attached to and under the control of the processor;

(c) a graphic device attached to and under the control of the processor;

(d) a modeling layer object in the storage;

(e) a grafport object in the storage;

(f) means for generating calls from the modeling layer object to the grafport object using a predefined set of graphic primitives;

(g) means for capturing state information and rendering information at the grafport object; and

(h) means for passing the state information and the rendering information to a graphic device object for output on the graphic device.

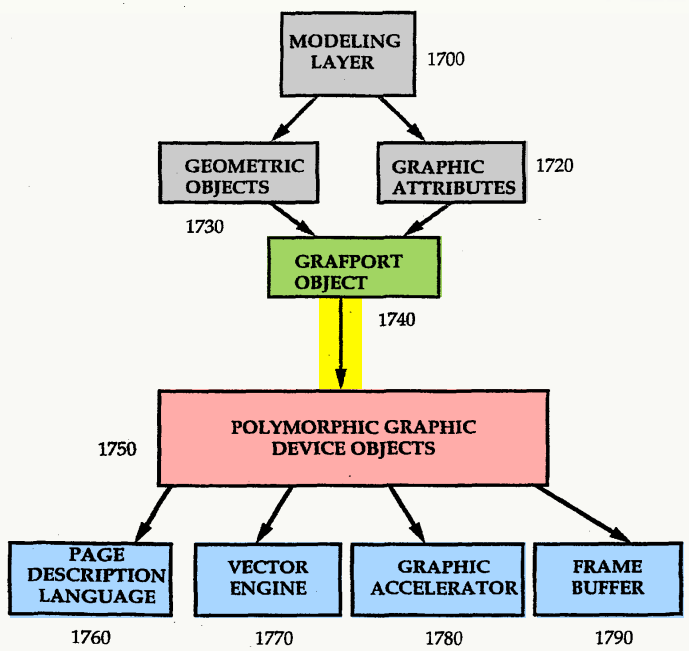


FIGURE 17

U.S. 5,455,599 (Grafport)

Android Code Infringing the '599 Patent

Graphic object:

instance of any subclass of:

android.graphics.drawable.shapes.Shape (rect, arc, oval, path, etc.)

or android.view.View (button, checkbox, listview, textview, etc.)

or android.graphics.drawable.Drawable (simple & composite object)

i.e. any class with a draw(Canvas) method

Grafport object:

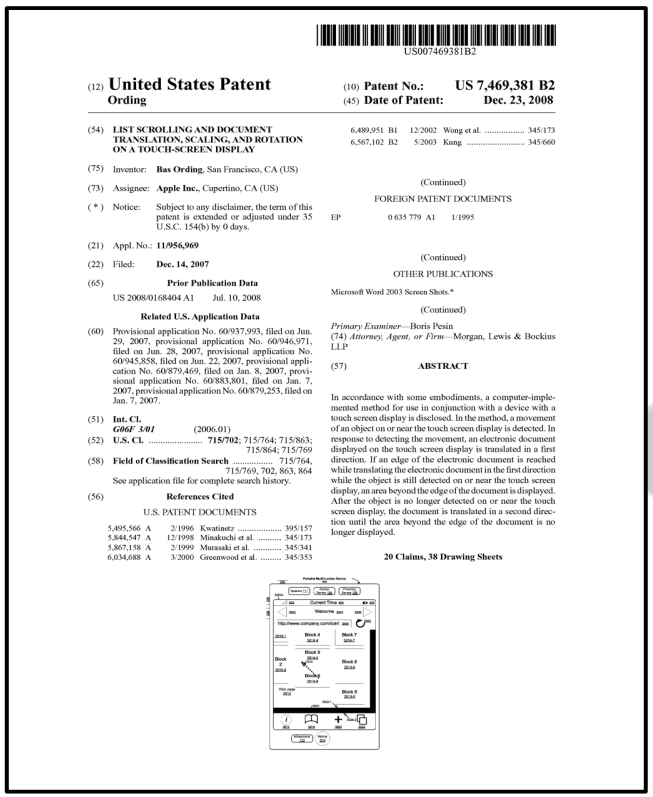
instance of android.graphics.Canvas

Graphic device object:

instance of any subclass of skia/src/core/SkDevice.cpp (e.g. SkGLDevice, etc.)

U.S. 7,469,381 (Bounce Scrolling)

- "List Scrolling And Document Translation, Scaling, And Rotation On A Touch-Screen Display"
- Filing Date: December 14, 2007
  - Priority: January 7, 2007



US007469381B2

(12) **United States Patent**
Ording

(10) **Patent No.: US 7,469,381 B2**
(45) **Date of Patent: Dec. 23, 2008**

(54) **LIST SCROLLING AND DOCUMENT TRANSLATION, SCALING, AND ROTATION ON A TOUCH-SCREEN DISPLAY**

(75) Inventor: **Bas Ording**, San Francisco, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **11/956,969**

(22) Filed: **Dec. 14, 2007**

(65) **Prior Publication Data**

US 2008/0168404 A1 Jul. 10, 2008

Related U.S. Application Data

(60) Provisional application No. 60/937,993, filed on Jun. 29, 2007, provisional application No. 60/946,971, filed on Jun. 28, 2007, provisional application No. 60/945,858, filed on Jun. 22, 2007, provisional application No. 60/879,469, filed on Jan. 8, 2007, provisional application No. 60/883,801, filed on Jan. 7, 2007, provisional application No. 60/879,253, filed on Jun. 7, 2007.

(51) **Int. Cl.**
G06F 3/01 (2006.01)

(52) **U.S. Cl.** **715/702**; 715/764; 715/863; 715/864; 715/769

(58) **Field of Classification Search** 715/764, 715/769, 702, 863, 864
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,495,566	A	2/1996	Kwatinetz	395/157
5,844,547	A	12/1998	Minakuchi et al.	345/173
5,867,158	A	2/1999	Murasaki et al.	345/341
6,034,688	A	3/2000	Greenwood et al.	345/353
6,489,951	B1	12/2002	Wong et al.	345/173
6,567,102	B2	5/2003	Kung	345/660

(Continued)

FOREIGN PATENT DOCUMENTS

EP 0 635 779 A1 1/1995

(Continued)

OTHER PUBLICATIONS

Microsoft Word 2003 Screen Shots.*

(Continued)

Primary Examiner—Boris Pesin
(74) *Attorney, Agent, or Firm*—Morgan, Lewis & Bockius LLP

(57) **ABSTRACT**

In accordance with some embodiments, a computer-implemented method for use in conjunction with a device with a touch screen display is disclosed. In the method, a movement of an object on or near the touch screen display is detected. In response to detecting the movement, an electronic document displayed on the touch screen display is translated in a first direction. If an edge of the electronic document is reached while translating the electronic document in the first direction while the object is still detected on or near the touch screen display, an area beyond the edge of the document is displayed. After the object is no longer detected on or near the touch screen display, the document is translated in a second direction until the area beyond the edge of the document is no longer displayed.

20 Claims, 38 Drawing Sheets

(57) **ABSTRACT**

In accordance with some embodiments, a computer-implemented method for use in conjunction with a device with a touch screen display is disclosed. In the method, a movement of an object on or near the touch screen display is detected. In response to detecting the movement, an electronic document displayed on the touch screen display is translated in a first direction. If an edge of the electronic document is reached while translating the electronic document in the first direction while the object is still detected on or near the touch screen display, an area beyond the edge of the document is displayed. After the object is no longer detected on or near the touch screen display, the document is translated in a second direction until the area beyond the edge of the document is no longer displayed.

U.S. 7,469,381 (Bounce Scrolling)

- '381 patent relates to bounce of scrolling view upon reaching end of content

Samsung Galaxy S Example

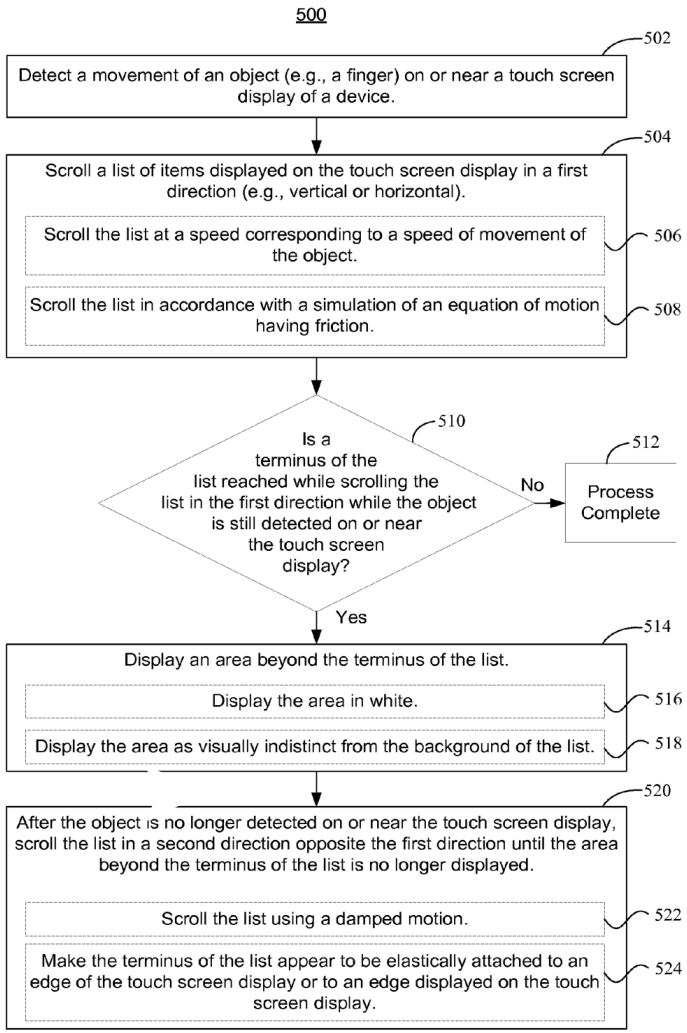
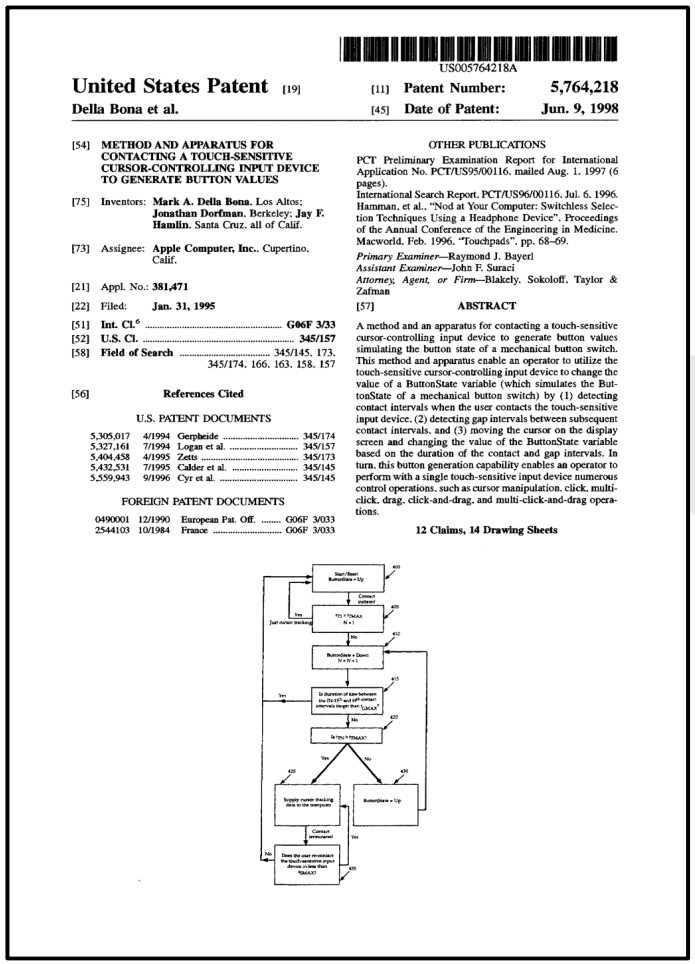


Figure 5

U.S. 5,764,218 (Gesture Detector)



US005764218A

United States Patent [19]
Della Bona et al.

[11] **Patent Number: 5,764,218**
[45] **Date of Patent: Jun. 9, 1998**

[54] **METHOD AND APPARATUS FOR CONTACTING A TOUCH-SENSITIVE CURSOR-CONTROLLING INPUT DEVICE TO GENERATE BUTTON VALUES**

[75] Inventors: **Mark A. Della Bona**, Los Altos; **Jonathan Dorfman**, Berkeley; **Jay F. Hamlin**, Santa Cruz, all of Calif.

[73] Assignee: **Apple Computer, Inc.**, Cupertino, Calif.

[21] Appl. No.: **381,471**

[22] Filed: **Jan. 31, 1995**

[51] **Int. Cl.**[6] **G06F 3/33**
[52] **U.S. Cl.** **345/157**
[58] **Field of Search** 345/145, 173, 345/174, 166, 163, 158, 157

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,305,017	4/1994	Gerpheide	345/174
5,327,161	7/1994	Logan et al.	345/157
5,404,458	4/1995	Zetts	345/173
5,432,531	7/1995	Calder et al.	345/145
5,559,943	9/1996	Cyr et al.	345/145

FOREIGN PATENT DOCUMENTS

0490001	12/1990	European Pat. Off.	G06F 3/033
2544103	10/1984	France	G06F 3/033

OTHER PUBLICATIONS

PCT Preliminary Examination Report for International Application No. PCT/US95/00116, mailed Aug. 1, 1997 (6 pages).
International Search Report, PCT/US96/00116, Jul. 6, 1996.
Hamman, et al., "Nod at Your Computer: Switchless Selection Techniques Using a Headphone Device", Proceedings of the Annual Conference of the Engineering in Medicine. Macworld, Feb. 1996, "Touchpads", pp. 68–69.

Primary Examiner—Raymond J. Bayerl
Assistant Examiner—John F. Suraci
Attorney, Agent, or Firm—Blakely, Sokoloff, Taylor & Zafman

[57] **ABSTRACT**

A method and an apparatus for contacting a touch-sensitive cursor-controlling input device to generate button values simulating the button state of a mechanical button switch. This method and apparatus enable an operator to utilize the touch-sensitive cursor-controlling input device to change the value of a ButtonState variable (which simulates the ButtonState of a mechanical button switch) by (1) detecting contact intervals when the user contacts the touch-sensitive input device, (2) detecting gap intervals between subsequent contact intervals, and (3) moving the cursor on the display screen and changing the value of the ButtonState variable based on the duration of the contact and gap intervals. In turn, this button generation capability enables an operator to perform with a single touch-sensitive input device numerous control operations, such as cursor manipulation, click, multi-click, drag, click-and-drag, and multi-click-and-drag operations.

12 Claims, 14 Drawing Sheets

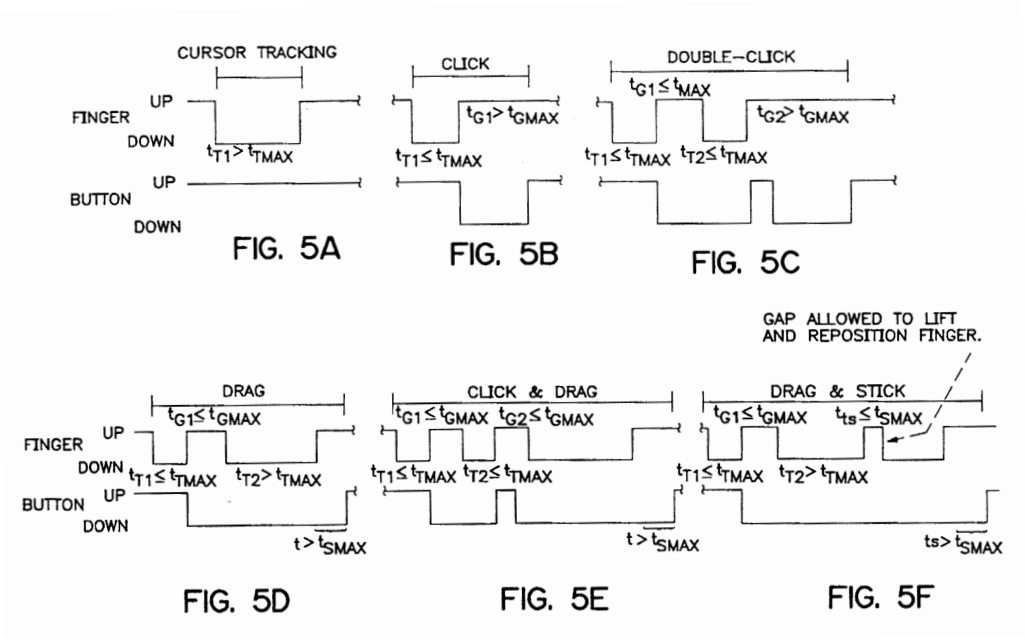
- "Method And Apparatus For Contacting A Touch-Sensitive Cursor-Controlling Input Device To Generate Button Values"
- Filing Date: January 31, 1995
  - Priority: March 3, 1993

[57] **ABSTRACT**

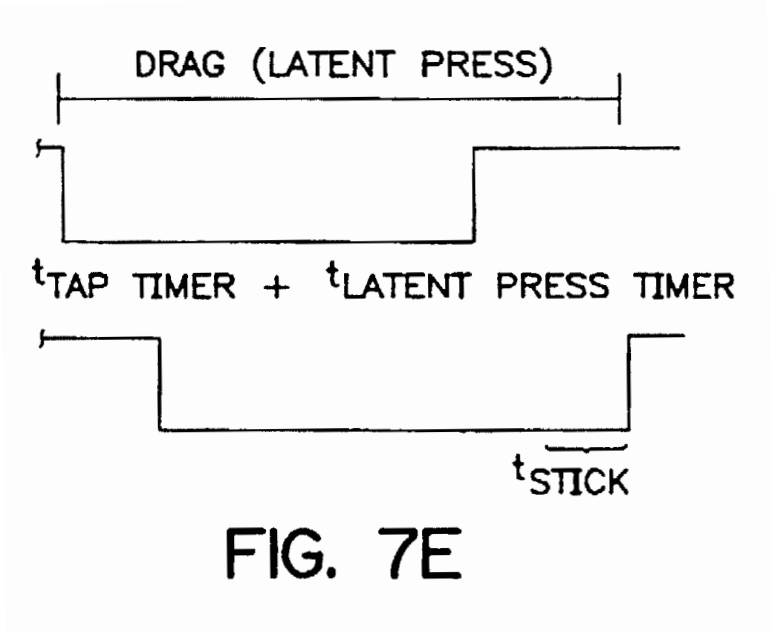
A method and an apparatus for contacting a touch-sensitive cursor-controlling input device to generate button values simulating the button state of a mechanical button switch. This method and apparatus enable an operator to utilize the touch-sensitive cursor-controlling input device to change the value of a ButtonState variable (which simulates the ButtonState of a mechanical button switch) by (1) detecting contact intervals when the user contacts the touch-sensitive input device, (2) detecting gap intervals between subsequent contact intervals, and (3) moving the cursor on the display screen and changing the value of the ButtonState variable based on the duration of the contact and gap intervals. In turn, this button generation capability enables an operator to perform with a single touch-sensitive input device numerous control operations, such as cursor manipulation, click, multi-click, drag, click-and-drag, and multi-click-and-drag operations.

U.S. 5,764,218 (Gesture Detector)

‘218 patent describes emulating mouse operations with a touch-sensitive input device



Gesture Examples



Long Press Example

U.S. 5,764,218 (Gesture Detector)

Android source code shows infringement of the '218 patent

public static class
GestureDetector.SimpleOnGestureListener

extends Object
implements GestureDetector.OnDoubleTapListener GestureDetector.OnGestureListener

java.lang.Object
↳android.view.GestureDetector.SimpleOnGestureListener

Class Overview

A convenience class to extend when you only want to listen for a subset of all the gestures. GestureDetector.OnDoubleTapListener but does nothing and return false for all a

package
android.gesture

Classes | Description

Provides classes to create, recognize, load and save gestures.

more...

Interfaces

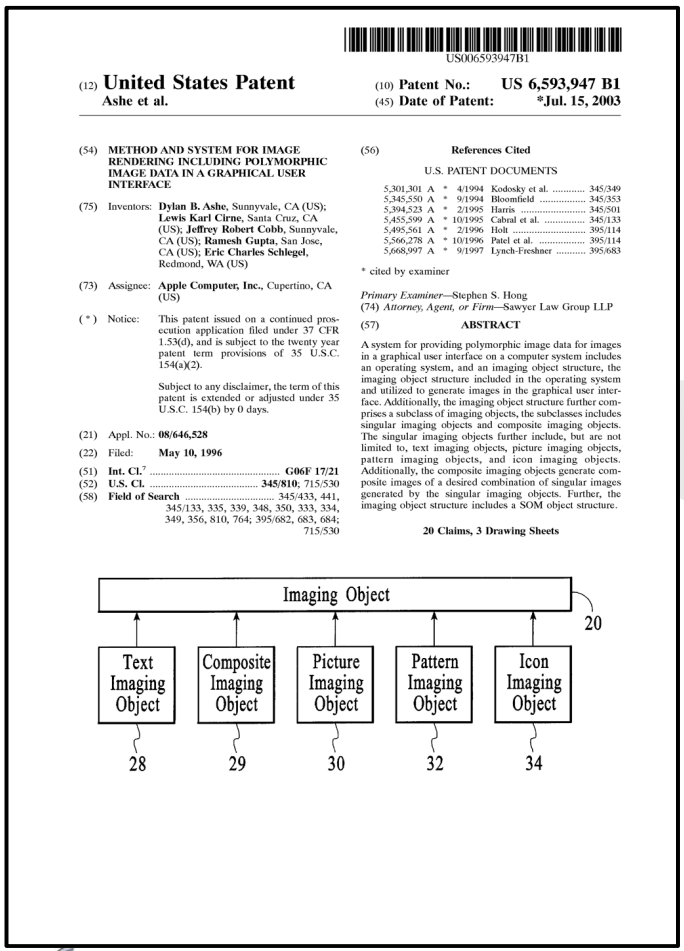
GestureOverlayView.OnGestureListener	
GestureOverlayView.OnGesturePerformedListener	
GestureOverlayView.OnGesturingListener	

Public Methods

boolean	onDoubleTap (MotionEvent e) Notified when a double-tap occurs.
boolean	onDoubleTapEvent (MotionEvent e) Notified when an event within a double-tap gesture occurs, including th
boolean	onDown (MotionEvent e) Notified when a tap occurs with the down MotionEvent that triggered it
boolean	onFling (MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) Notified of a fling event when it occurs with the initial on down MotionE
void	onLongPress (MotionEvent e) Notified when a long press occurs with the initial on down MotionEven
boolean	onScroll (MotionEvent e1, MotionEvent e2, float distanceX, float distanceY Notified when a scroll occurs with the initial on down MotionEvent and
void	onShowPress (MotionEvent e) The user has performed a down MotionEvent and not performed a mo
boolean	onSingleTapConfirmed (MotionEvent e) Notified when a single-tap occurs.
boolean	onSingleTapUp (MotionEvent e) Notified when a tap occurs with the up MotionEvent that triggered it.

U.S. 6,593,947 (Composite Objects)

- "Method and System For Image Rendering Including Polymorphic Image Data In A Graphical User Interface"
- Filing Date: May 10, 1996



US006593947B1

(12) **United States Patent**
Ashe et al.

(10) **Patent No.: US 6,593,947 B1**
(45) **Date of Patent: *Jul. 15, 2003**

(54) **METHOD AND SYSTEM FOR IMAGE RENDERING INCLUDING POLYMORPHIC IMAGE DATA IN A GRAPHICAL USER INTERFACE**

(75) Inventors: **Dylan B. Ashe**, Sunnyvale, CA (US); **Lewis Karl Cirne**, Santa Cruz, CA (US); **Jeffrey Robert Cobb**, Sunnyvale, CA (US); **Ramesh Gupta**, San Jose, CA (US); **Eric Charles Schlegel**, Redmond, WA (US)

(73) Assignee: **Apple Computer, Inc.**, Cupertino, CA (US)

(*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **08/646,528**

(22) Filed: **May 10, 1996**

(51) **Int. Cl.**[7] **G06F 17/21**
(52) **U.S. Cl.** **345/810**; 715/530
(58) **Field of Search** 345/433, 441, 345/133, 335, 339, 348, 350, 333, 334, 349, 356, 810, 764; 395/682, 683, 684; 715/530

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,301,301	A *	4/1994	Kodosky et al.	345/349
5,345,550	A *	9/1994	Bloomfield	345/353
5,394,523	A *	2/1995	Harris	345/501
5,455,599	A *	10/1995	Cabral et al.	345/133
5,495,561	A *	2/1996	Holt	395/114
5,566,278	A *	10/1996	Patel et al.	395/114
5,668,997	A *	9/1997	Lynch-Freshner	395/683

* cited by examiner

Primary Examiner—Stephen S. Hong
(74) *Attorney, Agent, or Firm*—Sawyer Law Group LLP

(57) **ABSTRACT**

A system for providing polymorphic image data for images in a graphical user interface on a computer system includes an operating system, and an imaging object structure, the imaging object structure included in the operating system and utilized to generate images in the graphical user interface. Additionally, the imaging object structure further comprises a subclass of imaging objects, the subclasses includes singular imaging objects and composite imaging objects. The singular imaging objects further include, but are not limited to, text imaging objects, picture imaging objects, pattern imaging objects, and icon imaging objects. Additionally, the composite imaging objects generate composite images of a desired combination of singular images generated by the singular imaging objects. Further, the imaging object structure includes a SOM object structure.

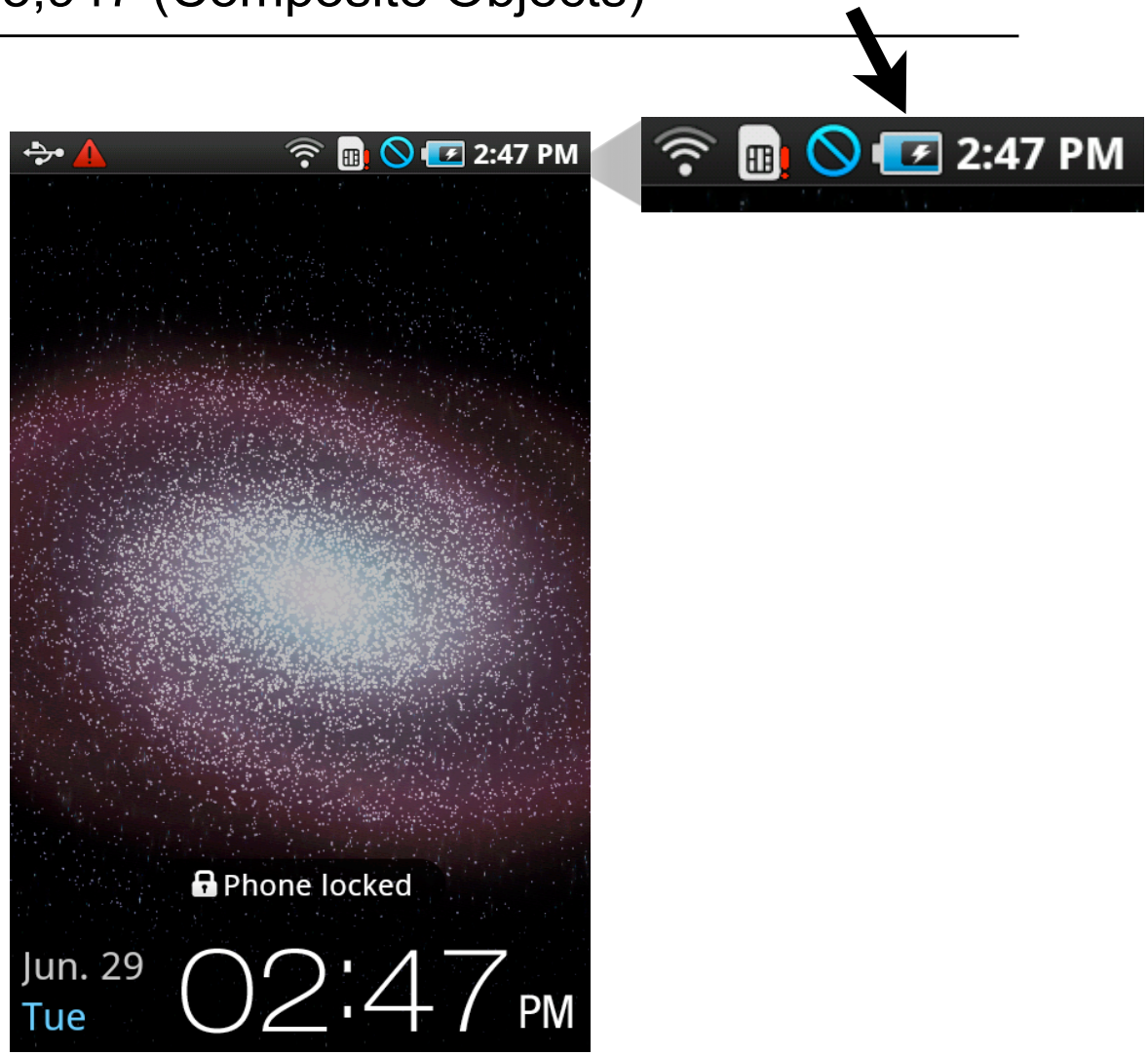
20 Claims, 3 Drawing Sheets

(57) **ABSTRACT**

A system for providing polymorphic image data for images in a graphical user interface on a computer system includes an operating system, and an imaging object structure, the imaging object structure included in the operating system and utilized to generate images in the graphical user interface. Additionally, the imaging object structure further comprises a subclass of imaging objects, the subclasses includes singular imaging objects and composite imaging objects. The singular imaging objects further include, but are not limited to, text imaging objects, picture imaging objects, pattern imaging objects, and icon imaging objects. Additionally, the composite imaging objects generate composite images of a desired combination of singular images generated by the singular imaging objects. Further, the imaging object structure includes a SOM object structure.

U.S. 6,593,947 (Composite Objects)

- '947 describes a container for managing a collection of imaging objects
- One example is a battery level indicator associated with an array of images with corresponding numbers
- When a specified value is reached there is a display of the corresponding image

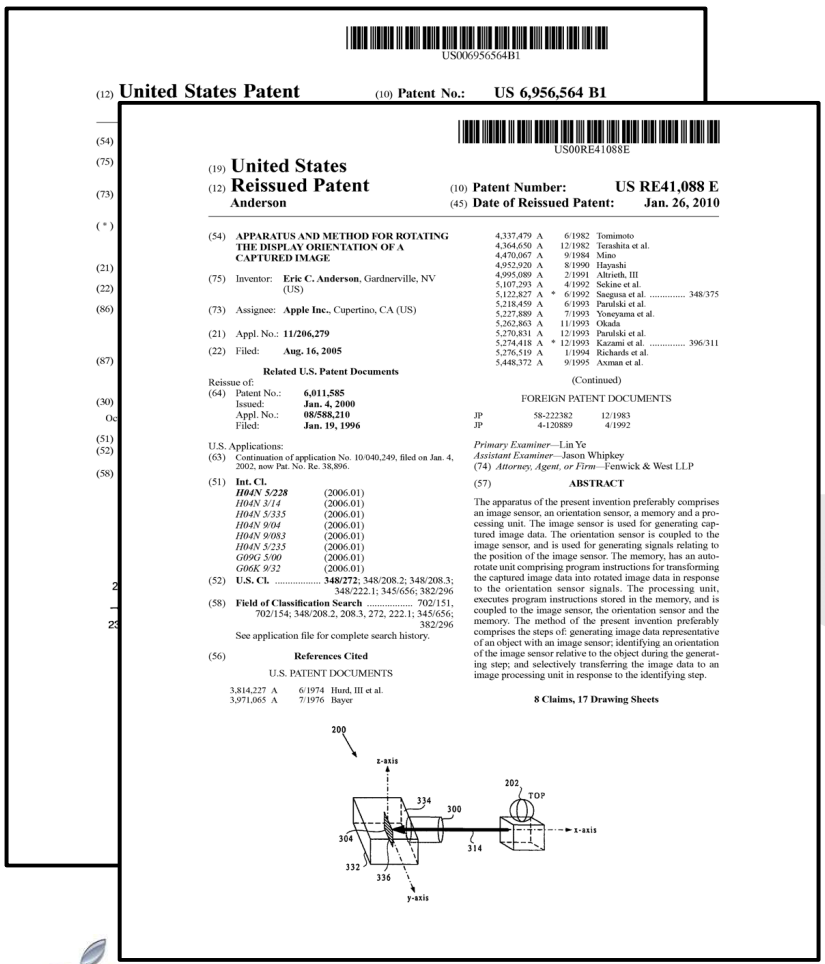


U.S. 6,593,947 (Composite Objects)

- Reads on subclasses of android.graphics.drawable.Drawable
- Android has both superimposing-style and selecting-style "composite" imaging object subclasses:
  - LayerDrawable - superimpose a set of images
  - TransitionDrawable - cross-fade between pair of images
  - AnimationDrawable - display images successively, like a movie
  - NinePatchDrawable - draw scalable button using side and corner images
  - LevelListDrawable - select from list of images (e.g. battery level indicator)
  - StateListDrawable - select among images for enabled, selected, pressed, etc.
- Android also has "singular" imaging object subclasses:
  - BitmapDrawble - draw a bitmap
  - ColorDrawable - draw a region filled with a color
  - GradientDrawable - draw a region filled with a gradient
  - PictureDrawble - playback any sequence of drawing commands
  - ShapeDrawable - draw a graphic primitive (arc, oval, rect, round rect, path, ...)

U.S. RE41,088 & U.S. 6,956,564 (Rotate Display)

- "Apparatus And Method For Rotating The Display Orientation Of A Captured Image"
- Filing Date: January 19, 1996



US006956564B1

(12) **United States Patent** (10) **Patent No.: US 6,956,564 B1**

US00RE41088E

(19) **United States**
(12) **Reissued Patent**
Anderson

(10) **Patent Number: US RE41,088 E**
(45) **Date of Reissued Patent: Jan. 26, 2010**

(54) **APPARATUS AND METHOD FOR ROTATING THE DISPLAY ORIENTATION OF A CAPTURED IMAGE**

(75) Inventor: **Eric C. Anderson**, Gardnerville, NV (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(21) Appl. No.: **11/206,279**

(22) Filed: **Aug. 16, 2005**

Related U.S. Patent Documents

Reissue of:
(64) Patent No.: **6,011,585**
Issued: **Jan. 4, 2000**
Appl. No.: **08/588,210**
Filed: **Jan. 19, 1996**

U.S. Applications:
(63) Continuation of application No. 10/040,249, filed on Jan. 4, 2002, now Pat. No. Re. 38,896.

(51) **Int. Cl.**
H04N 5/228 (2006.01)
H04N 3/14 (2006.01)
H04N 5/335 (2006.01)
H04N 9/04 (2006.01)
H04N 9/083 (2006.01)
H04N 5/235 (2006.01)
G09G 5/00 (2006.01)
G06K 9/32 (2006.01)

(52) **U.S. Cl.** **348/272**; 348/208.2; 348/208.3; 348/222.1; 345/656; 382/296

(58) **Field of Classification Search** 702/151; 702/154; 348/208.2, 208.3, 272, 222.1; 345/656; 382/296
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,814,227	A	6/1974	Hurd, III et al.	
3,971,065	A	7/1976	Bayer	
4,337,479	A	6/1982	Tomimoto	
4,364,650	A	12/1982	Terashita et al.	
4,470,067	A	9/1984	Mino	
4,952,920	A	8/1990	Hayashi	
4,995,089	A	2/1991	Altrieth, III	
5,107,293	A	4/1992	Sekine et al.	
5,122,827	A *	6/1992	Saegusa et al.	348/375
5,218,459	A	6/1993	Parulski et al.	
5,227,889	A	7/1993	Yoneyama et al.	
5,262,863	A	11/1993	Okada	
5,270,831	A	12/1993	Parulski et al.	
5,274,418	A *	12/1993	Kazami et al.	396/311
5,276,519	A	1/1994	Richards et al.	
5,448,372	A	9/1995	Axman et al.	

(Continued)

FOREIGN PATENT DOCUMENTS

JP	58-222382	12/1983
JP	4-120889	4/1992

Primary Examiner—Lin Ye
Assistant Examiner—Jason Whipkey
(74) *Attorney, Agent, or Firm*—Fenwick & West LLP

(57) **ABSTRACT**

The apparatus of the present invention preferably comprises an image sensor, an orientation sensor, a memory and a processing unit. The image sensor is used for generating captured image data. The orientation sensor is coupled to the image sensor, and is used for generating signals relating to the position of the image sensor. The memory, has an auto-rotate unit comprising program instructions for transforming the captured image data into rotated image data in response to the orientation sensor signals. The processing unit, executes program instructions stored in the memory, and is coupled to the image sensor, the orientation sensor and the memory. The method of the present invention preferably comprises the steps of: generating image data representative of an object with an image sensor; identifying an orientation of the image sensor relative to the object during the generating step; and selectively transferring the image data to an image processing unit in response to the identifying step.

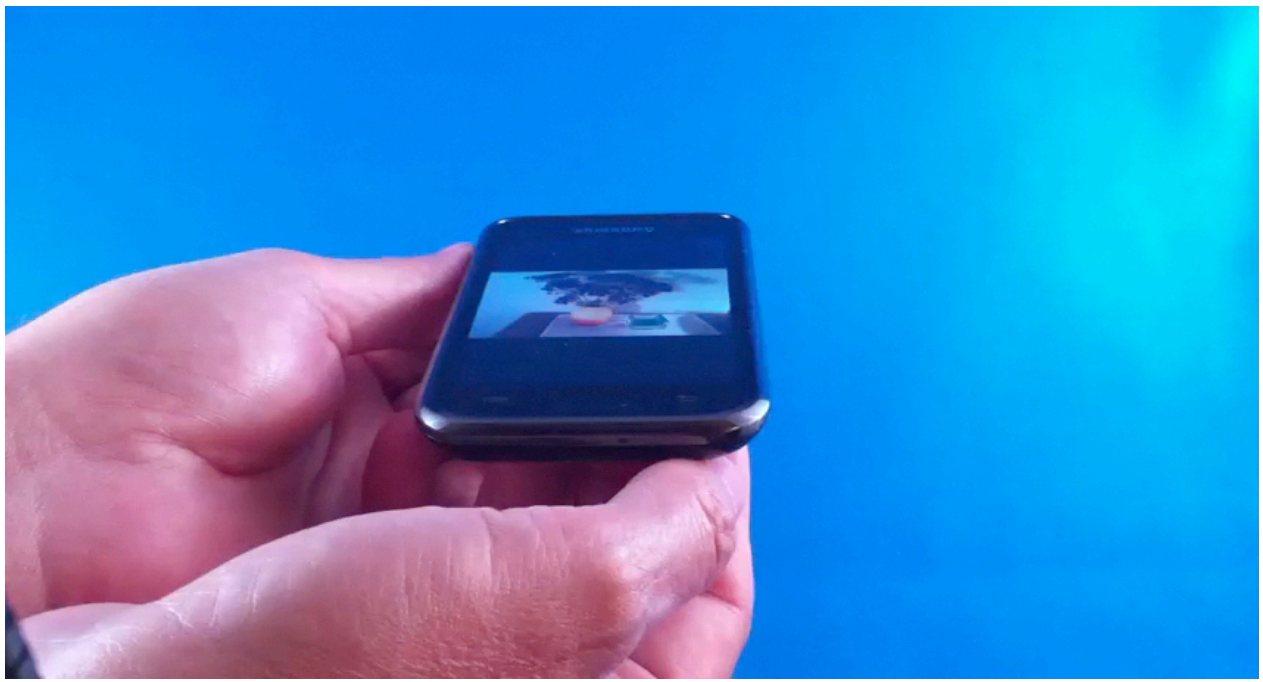
8 Claims, 17 Drawing Sheets

(57) **ABSTRACT**

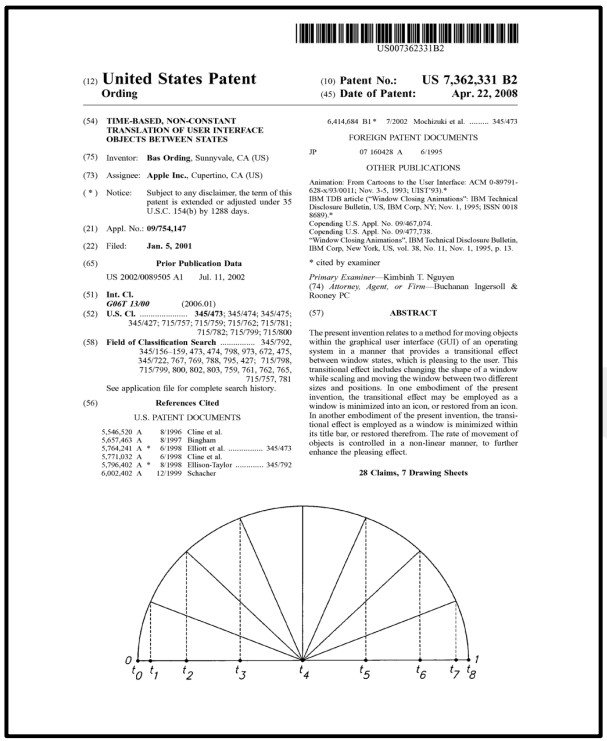
The apparatus of the present invention preferably comprises an image sensor, an orientation sensor, a memory and a processing unit. The image sensor is used for generating captured image data. The orientation sensor is coupled to the image sensor, and is used for generating signals relating to the position of the image sensor. The memory, has an auto-rotate unit comprising program instructions for transforming the captured image data into rotated image data in response to the orientation sensor signals. The processing unit, executes program instructions stored in the memory, and is coupled to the image sensor, the orientation sensor and the memory. The method of the present invention preferably comprises the steps of: generating image data representative of an object with an image sensor; identifying an orientation of the image sensor relative to the object during the generating step; and selectively transferring the image data to an image processing unit in response to the identifying step.

U.S. RE41,088 & U.S. 6,956,564 (Rotate Display)

- '088 and '564 describe rotating a display orientation of an image based on device position
- '088 further includes rotating a display based on device pitch and roll



U.S. 7,362,331 (Non-Linear Animation)



US007362331B2

(12) **United States Patent**
Ording

(10) **Patent No.: US 7,362,331 B2**
(45) **Date of Patent: Apr. 22, 2008**

(54) **TIME-BASED, NON-CONSTANT TRANSLATION OF USER INTERFACE OBJECTS BETWEEN STATES**

(75) Inventor: **Bas Ording**, Sunnyvale, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1288 days.

(21) Appl. No.: **09/754,147**

(22) Filed: **Jan. 5, 2001**

(65) **Prior Publication Data**

US 2002/0089505 A1 Jul. 11, 2002

(51) **Int. Cl.**
G06T 13/00 (2006.01)

(52) **U.S. Cl.** **345/473**; 345/474; 345/475; 345/427; 715/757; 715/759; 715/762; 715/781; 715/782; 715/799; 715/800

(58) **Field of Classification Search** 345/792, 345/156–159, 473, 474, 798, 973, 672, 475, 345/722, 767, 769, 788, 795, 427; 715/798, 715/799, 800, 802, 803, 759, 761, 762, 765, 715/757, 781

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,546,520 A 8/1996 Cline et al.
5,657,463 A 8/1997 Bingham
5,764,241 A * 6/1998 Elliott et al. 345/473
5,771,032 A 6/1998 Cline et al.
5,796,402 A * 8/1998 Ellison-Taylor 345/792
6,002,402 A 12/1999 Schacher
6,414,684 B1 * 7/2002 Mochizuki et al. 345/473

FOREIGN PATENT DOCUMENTS

JP 07 160428 A 6/1995

OTHER PUBLICATIONS

Animation: From Cartoons to the User Interface: ACM 0-89791-628-x/93/0011; Nov. 3-5, 1993, UIST'93).*
IBM TDB article ("Window Closing Animations": IBM Technical Disclosure Bulletin, US, IBM Corp, NY, Nov. 1, 1995; ISSN 0018 8689).*
Copending U.S. Appl. No. 09/467,074.
Copending U.S. Appl. No. 09/477,738.
"Window Closing Animations", IBM Technical Disclosure Bulletin, IBM Corp. New York, US, vol. 38, No. 11, Nov. 1, 1995, p. 13.

* cited by examiner

Primary Examiner—Kimbinh T. Nguyen
(74) *Attorney, Agent, or Firm*—Buchanan Ingersoll & Rooney PC

(57) **ABSTRACT**

The present invention relates to a method for moving objects within the graphical user interface (GUI) of an operating system in a manner that provides a transitional effect between window states, which is pleasing to the user. This transitional effect includes changing the shape of a window while scaling and moving the window between two different sizes and positions. In one embodiment of the present invention, the transitional effect may be employed as a window is minimized into an icon, or restored from an icon. In another embodiment of the present invention, the transitional effect is employed as a window is minimized within its title bar, or restored therefrom. The rate of movement of objects is controlled in a non-linear manner, to further enhance the pleasing effect.

28 Claims, 7 Drawing Sheets

- "Time-Based, Non-Constant Translation Of User Interface Objects Between States"
- Filing Date: January 5, 2001

(57) **ABSTRACT**

The present invention relates to a method for moving objects within the graphical user interface (GUI) of an operating system in a manner that provides a transitional effect between window states, which is pleasing to the user. This transitional effect includes changing the shape of a window while scaling and moving the window between two different sizes and positions. In one embodiment of the present invention, the transitional effect may be employed as a window is minimized into an icon, or restored from an icon. In another embodiment of the present invention, the transitional effect is employed as a window is minimized within its title bar, or restored therefrom. The rate of movement of objects is controlled in a non-linear manner, to further enhance the pleasing effect.

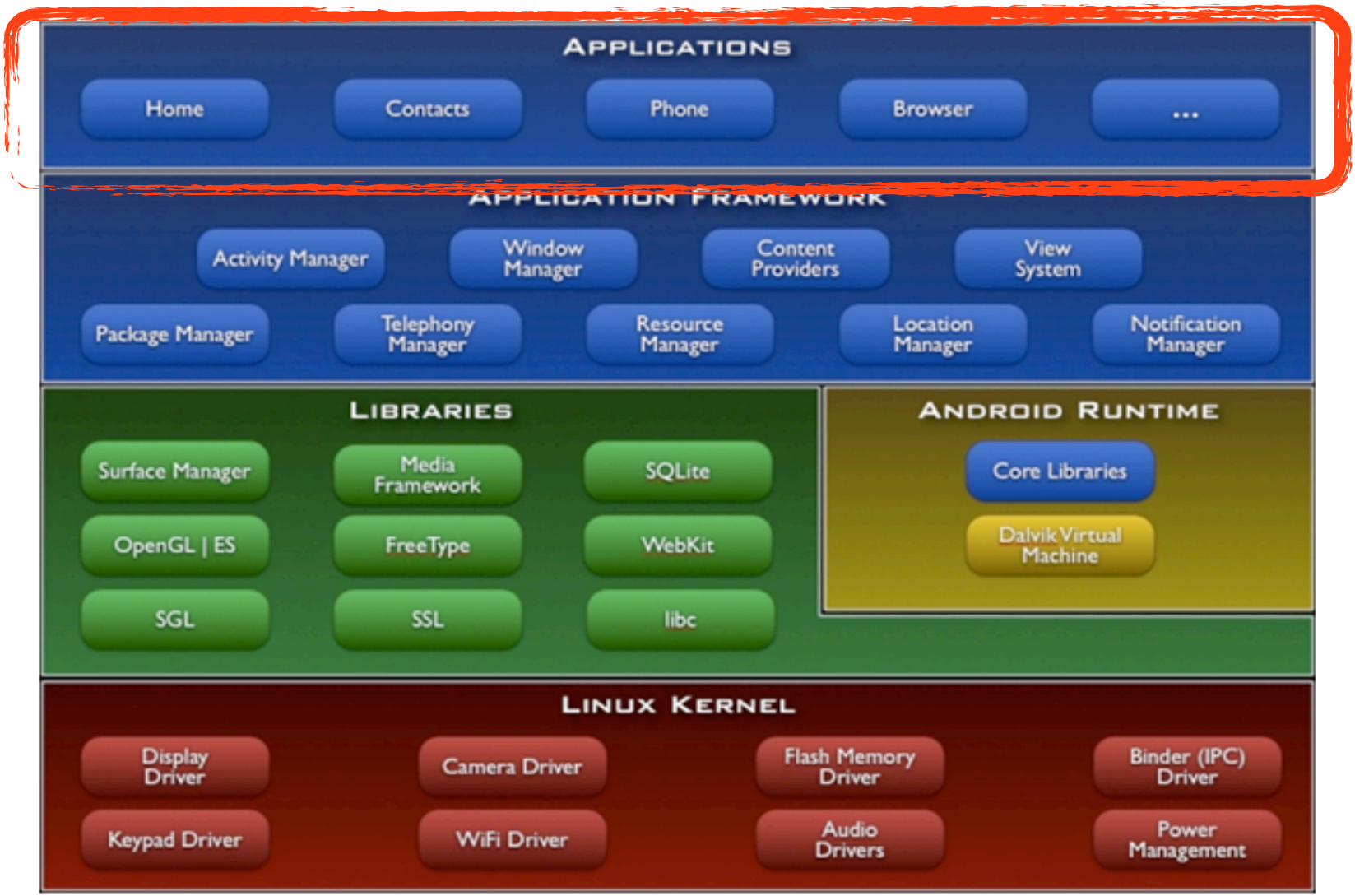
U.S. 7,362,331 (Non-Linear Animation)

- '331 relates to a method for moving objects within the GUI of an operating system in a manner that provides a transitional effect
- '331 provides for animation of user interface including ease in, ease out, bounce, etc.
- android.view.animation

'331 Claim 1

1. A method for moving an object in a graphical user interface, comprising the steps of:
   a) determining a path of movement for the object along at least one axis, and a period of time for the movement along said path;
   b) establishing a non-constant velocity function along said axis for said period of time;
   c) calculating an instantaneous position for the object along said path in accordance with said function and the relationship of a current time value to said period of time;
   d) displaying said object at said calculated position; and
   e) iteratively repeating steps (c) and (d) during said period of time.

Android Stack



Highlights of Example Apple Patents Infringed by Android Applications Layer

U.S. 7,479,949 (Gesture Heuristics)

U.S. 5,946,647 (Data Detector)

U.S. 7,657,849 (Unlock Gesture)

U.S. U.S. 6,072,489 & 5,949,432 (Translucent GUI)

U.S. 7,602,378 (Switchable Soft Keyboard)

U.S. 6,236,396 (Calendar Scheduler)

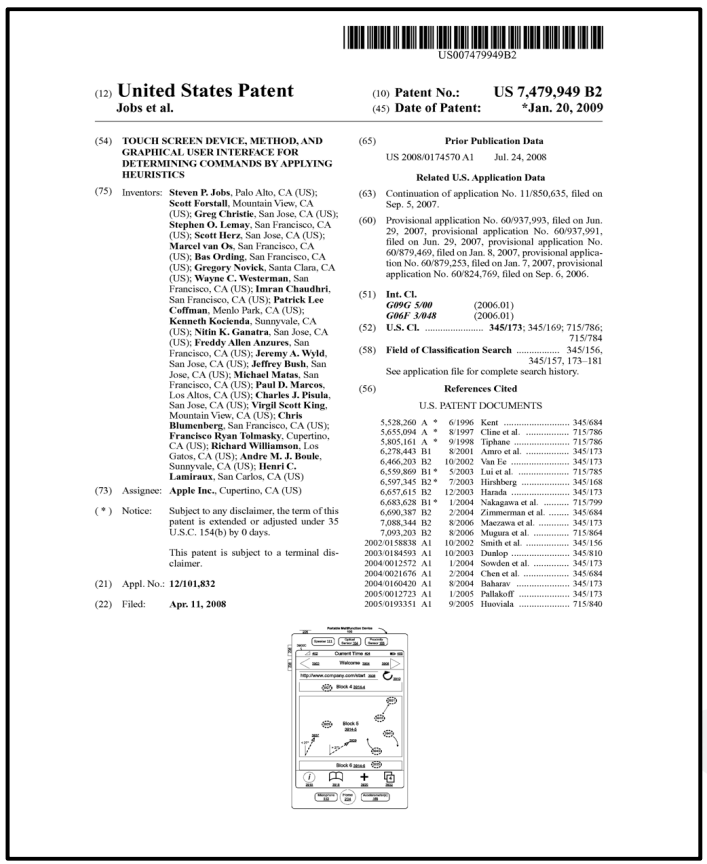
U.S. 7,669,134 (Messaging UI)

U.S. 5,544,358 & 5,446,882 (Address Book UI)

U.S. 6,493,002 (Status Bar)

U.S. 7,479,949 (Gesture Heuristics)

- "Touch Screen Device, Method, And Graphical User Interface For Determining Commands By Applying Heuristics"
- Filing Date: April 11, 2008
  - Priority: September 6, 2006



US007479949B2

(12) **United States Patent**
Jobs et al.

(10) **Patent No.: US 7,479,949 B2**
(45) **Date of Patent: *Jan. 20, 2009**

(54) **TOUCH SCREEN DEVICE, METHOD, AND GRAPHICAL USER INTERFACE FOR DETERMINING COMMANDS BY APPLYING HEURISTICS**

(75) Inventors: **Steven P. Jobs**, Palo Alto, CA (US); **Scott Forstall**, Mountain View, CA (US); **Greg Christie**, San Jose, CA (US); **Stephen O. Lemay**, San Francisco, CA (US); **Scott Herz**, San Jose, CA (US); **Marcel van Os**, San Francisco, CA (US); **Bas Ording**, San Francisco, CA (US); **Gregory Novick**, Santa Clara, CA (US); **Wayne C. Westerman**, San Francisco, CA (US); **Imran Chaudhri**, San Francisco, CA (US); **Patrick Lee Coffman**, Menlo Park, CA (US); **Kenneth Kocienda**, Sunnyvale, CA (US); **Nitin K. Ganatra**, San Jose, CA (US); **Freddy Allen Anzures**, San Francisco, CA (US); **Jeremy A. Wyld**, San Jose, CA (US); **Jeffrey Bush**, San Jose, CA (US); **Michael Matas**, San Francisco, CA (US); **Paul D. Marcos**, Los Altos, CA (US); **Charles J. Pisula**, San Jose, CA (US); **Virgil Scott King**, Mountain View, CA (US); **Chris Blumenberg**, San Francisco, CA (US); **Francisco Ryan Tolmasky**, Cupertino, CA (US); **Richard Williamson**, Los Gatos, CA (US); **Andre M. J. Boule**, Sunnyvale, CA (US); **Henri C. Lamiraux**, San Carlos, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **12/101,832**

(22) Filed: **Apr. 11, 2008**

(65) **Prior Publication Data**

US 2008/0174570 A1 Jul. 24, 2008

Related U.S. Application Data

(63) Continuation of application No. 11/850,635, filed on Sep. 5, 2007.

(60) Provisional application No. 60/937,993, filed on Jun. 29, 2007, provisional application No. 60/937,991, filed on Jun. 29, 2007, provisional application No. 60/879,469, filed on Jan. 8, 2007, provisional application No. 60/879,253, filed on Jan. 7, 2007, provisional application No. 60/824,769, filed on Sep. 6, 2006.

(51) **Int. Cl.**
G09G 5/00 (2006.01)
G06F 3/048 (2006.01)

(52) **U.S. Cl.** **345/173**; 345/169; 715/786; 715/784

(58) **Field of Classification Search** 345/156, 345/157, 173–181
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,528,260	A *	6/1996	Kent	345/684
5,555,094	A *	8/1997	Cline et al.	715/786
5,805,161	A *	9/1998	Tiphane	715/786
6,278,443	B1	8/2001	Amro et al.	345/173
6,466,203	B2	10/2002	Van Ee	345/173
6,559,869	B1 *	5/2003	Lui et al.	715/785
6,597,345	B2 *	7/2003	Hirshberg	345/168
6,657,615	B2	12/2003	Harada	345/173
6,683,628	B1 *	1/2004	Nakagawa et al.	715/799
6,690,387	B2	2/2004	Zimmerman et al.	345/684
7,088,344	B2	8/2006	Maezawa et al.	345/173
7,093,203	B2	8/2006	Mugura et al.	715/864
2002/0158838	A1	10/2002	Smith et al.	345/156
2003/0184593	A1	10/2003	Dunlop	345/810
2004/0012572	A1	1/2004	Sowden et al.	345/173
2004/0021676	A1	2/2004	Chen et al.	345/684
2004/0160420	A1	8/2004	Baharav	345/173
2005/0012723	A1	1/2005	Pallakoff	345/173
2005/0193351	A1	9/2005	Huoviala	715/840

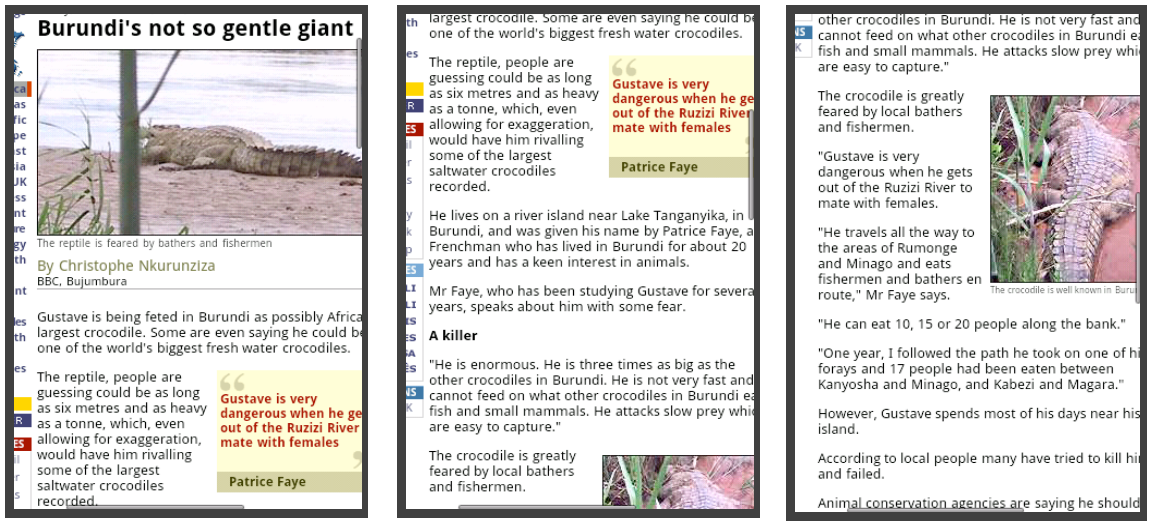
(57) **ABSTRACT**

A computer-implemented method for use in conjunction with a computing device with a touch screen display comprises: detecting one or more finger contacts with the touch screen display, applying one or more heuristics to the one or more finger contacts to determine a command for the device, and processing the command. The one or more heuristics comprise: a heuristic for determining that the one or more finger contacts correspond to a one-dimensional vertical screen scrolling command, a heuristic for determining that the one or more finger contacts correspond to a two-dimensional screen translation command, and a heuristic for determining that the one or more finger contacts correspond to a command to transition from displaying a respective item in a set of items to displaying a next item in the set of items.

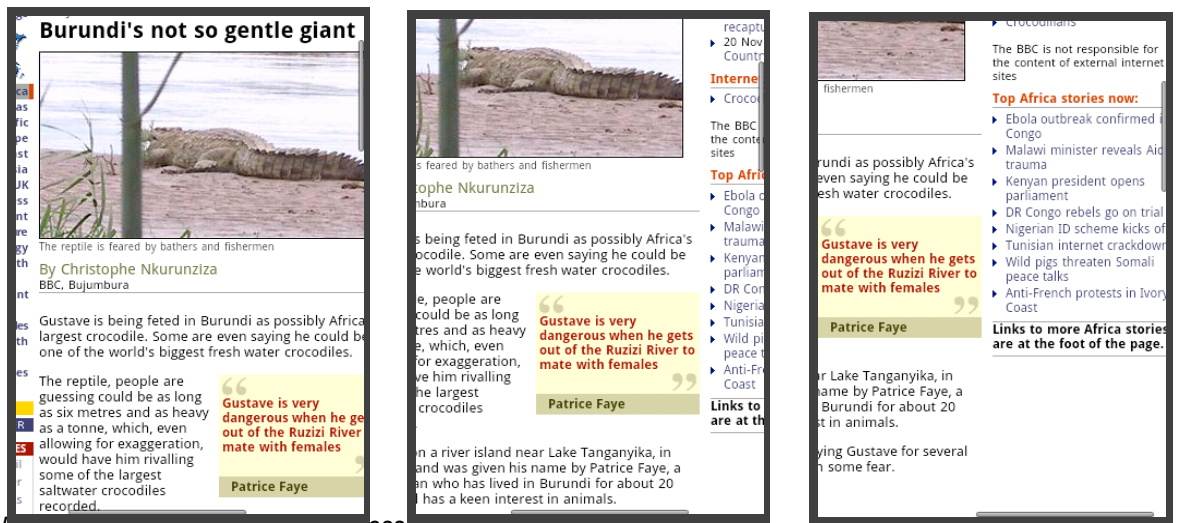
U.S. 7,479,949 (Gesture Heuristics)

‘949 patent discloses distinguishing 2D panning versus 1D scrolling touch gestures by detecting how close a movement is to a horizontal gesture as compared with a vertical gesture

One-Dimensional Vertical Scrolling in Web Browser



Two-Dimensional Vertical Panning in Web Browser

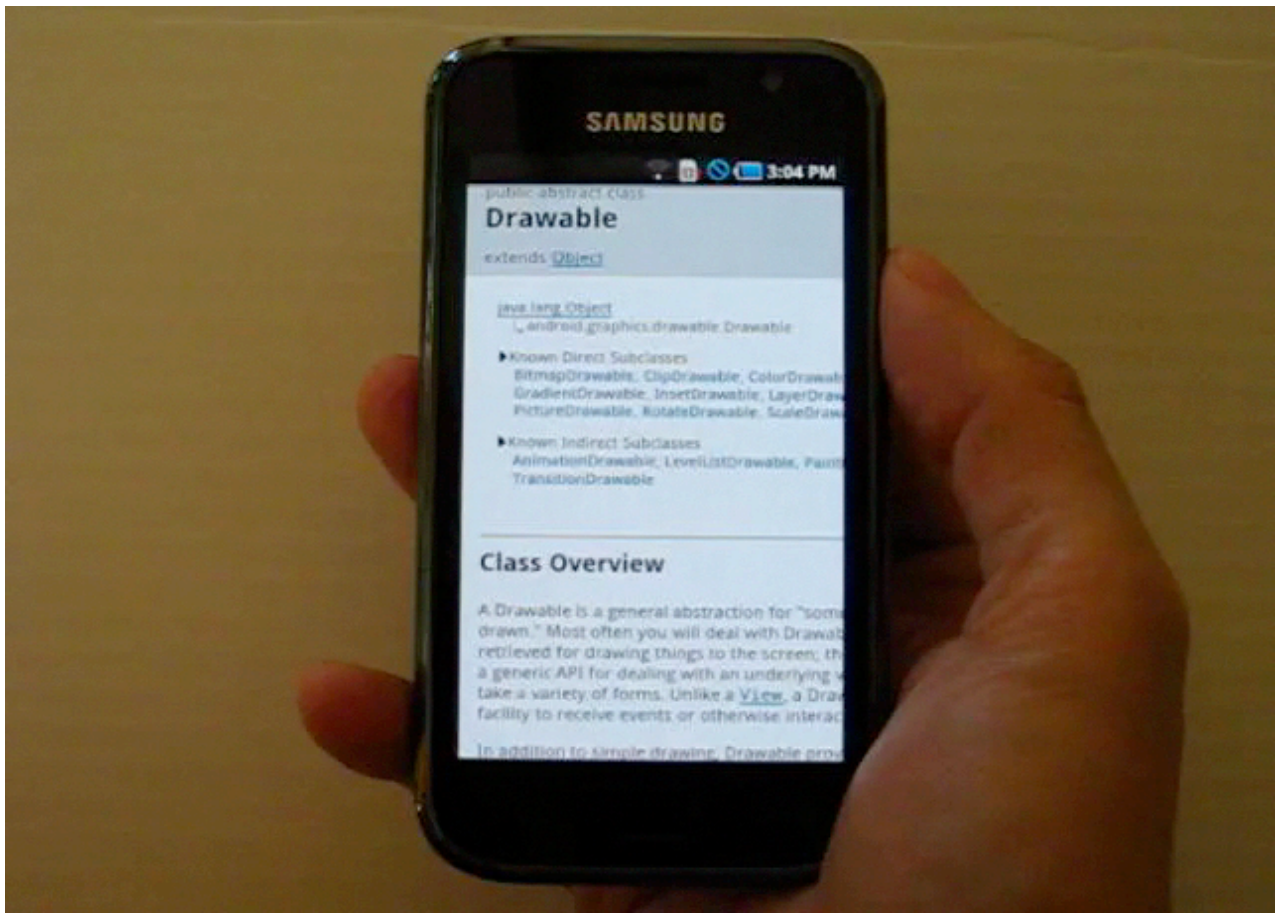


Slide your finger left or right to scroll horizontally.

Slide your finger up or down to scroll vertically.

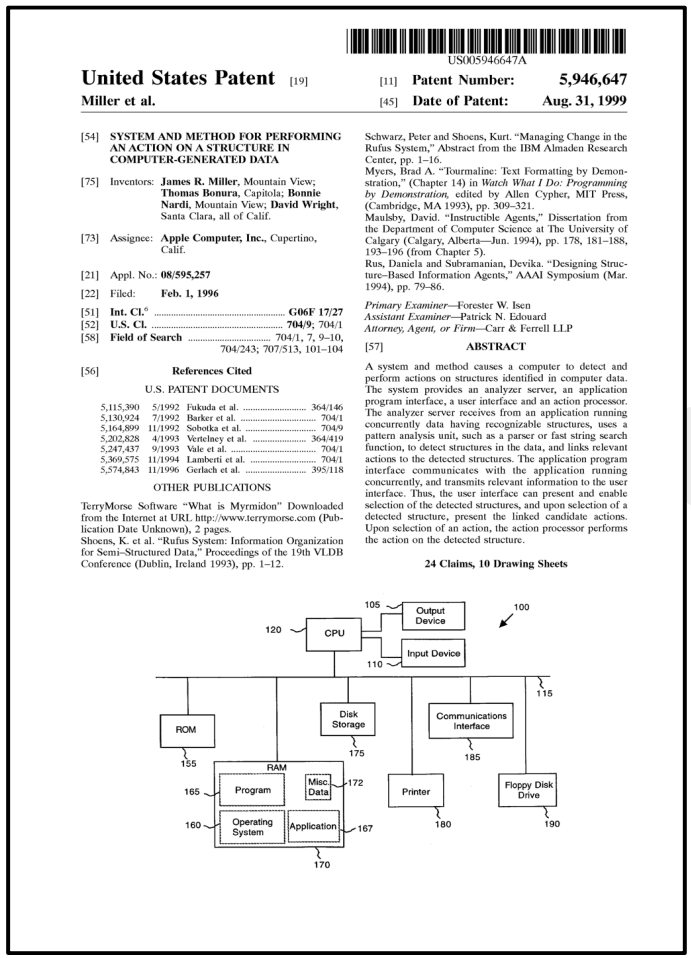
U.S. 7,479,949 (Gesture Heuristics)

Samsung Galaxy S Example



U.S. 5,946,647 (Data Detector)

- "System And Method For Performing An Action On A Structure In Computer-Generated Data"
- Filing Date: February 1, 1996



US005946647A

United States Patent [19]
Miller et al.

[11] **Patent Number: 5,946,647**
[45] **Date of Patent: Aug. 31, 1999**

[54] **SYSTEM AND METHOD FOR PERFORMING AN ACTION ON A STRUCTURE IN COMPUTER-GENERATED DATA**

[75] Inventors: **James R. Miller**, Mountain View; **Thomas Bonura**, Capitola; **Bonnie Nardi**, Mountain View; **David Wright**, Santa Clara, all of Calif.

[73] Assignee: **Apple Computer, Inc.**, Cupertino, Calif.

[21] Appl. No.: **08/595,257**

[22] Filed: **Feb. 1, 1996**

[51] **Int. Cl.**[6] **G06F 17/27**
[52] **U.S. Cl.** **704/9**; 704/1
[58] **Field of Search** 704/1, 7, 9–10, 704/243; 707/513, 101–104

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,115,390	5/1992	Fukuda et al.	364/146
5,130,924	7/1992	Barker et al.	704/1
5,164,899	11/1992	Sobotka et al.	704/9
5,202,828	4/1993	Vertelney et al.	364/419
5,247,437	9/1993	Vale et al.	704/1
5,369,575	11/1994	Lamberti et al.	704/1
5,574,843	11/1996	Gerlach et al.	395/118

OTHER PUBLICATIONS

TerryMorse Software "What is Myrmidon" Downloaded from the Internet at URL http://www.terrymorse.com (Publication Date Unknown), 2 pages.
Shoens, K. et al. "Rufus System: Information Organization for Semi-Structured Data," Proceedings of the 19th VLDB Conference (Dublin, Ireland 1993), pp. 1–12.
Schwarz, Peter and Shoens, Kurt. "Managing Change in the Rufus System," Abstract from the IBM Almaden Research Center, pp. 1–16.
Myers, Brad A. "Tourmaline: Text Formatting by Demonstration," (Chapter 14) in *Watch What I Do: Programming by Demonstration,* edited by Allen Cypher, MIT Press, (Cambridge, MA 1993), pp. 309–321.
Maulsby, David. "Instructible Agents," Dissertation from the Department of Computer Science at The University of Calgary (Calgary, Alberta—Jun. 1994), pp. 178, 181–188, 193–196 (from Chapter 5).
Rus, Daniela and Subramanian, Devika. "Designing Structure–Based Information Agents," AAAI Symposium (Mar. 1994), pp. 79–86.

Primary Examiner—Forester W. Isen
Assistant Examiner—Patrick N. Edouard
Attorney, Agent, or Firm—Carr & Ferrell LLP

[57] **ABSTRACT**

A system and method causes a computer to detect and perform actions on structures identified in computer data. The system provides an analyzer server, an application program interface, a user interface and an action processor. The analyzer server receives from an application running concurrently data having recognizable structures, uses a pattern analysis unit, such as a parser or fast string search function, to detect structures in the data, and links relevant actions to the detected structures. The application program interface communicates with the application running concurrently, and transmits relevant information to the user interface. Thus, the user interface can present and enable selection of the detected structures, and upon selection of a detected structure, present the linked candidate actions. Upon selection of an action, the action processor performs the action on the detected structure.

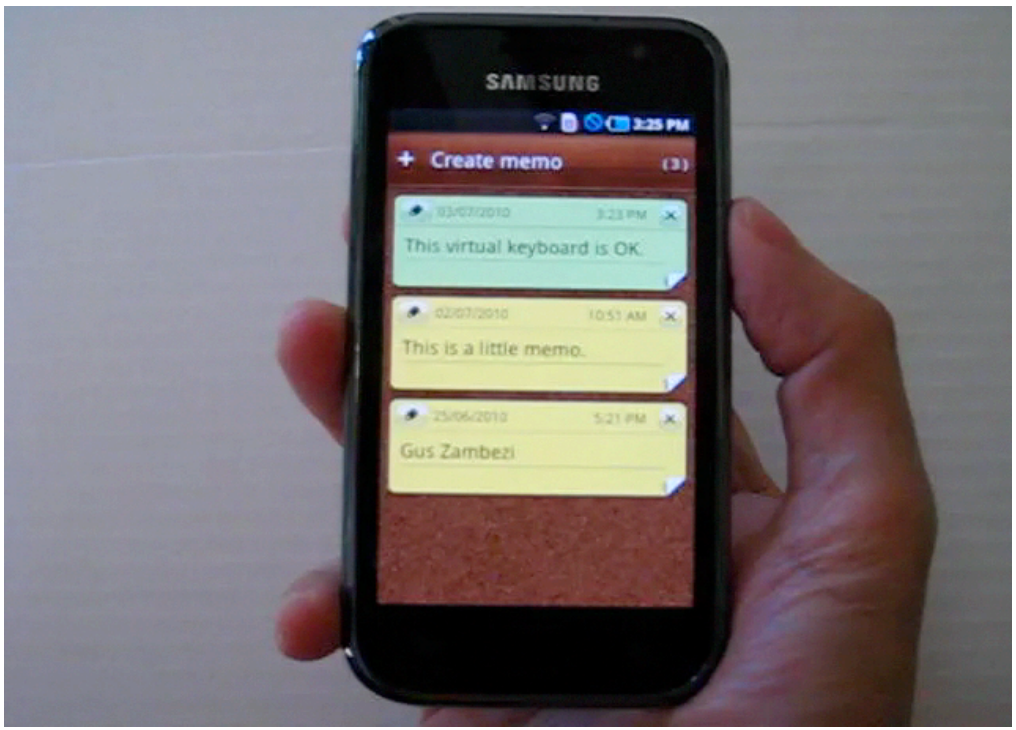
24 Claims, 10 Drawing Sheets

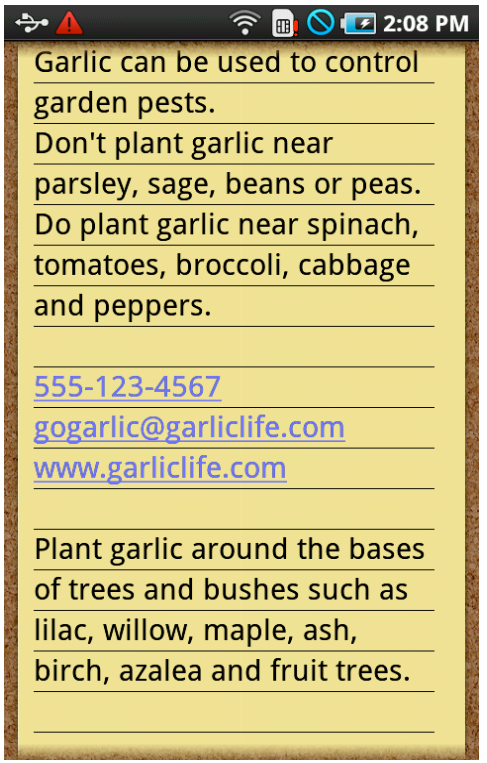
[57] **ABSTRACT**

A system and method causes a computer to detect and perform actions on structures identified in computer data. The system provides an analyzer server, an application program interface, a user interface and an action processor. The analyzer server receives from an application running concurrently data having recognizable structures, uses a pattern analysis unit, such as a parser or fast string search function, to detect structures in the data, and links relevant actions to the detected structures. The application program interface communicates with the application running concurrently, and transmits relevant information to the user interface. Thus, the user interface can present and enable selection of the detected structures, and upon selection of a detected structure, present the linked candidate actions. Upon selection of an action, the action processor performs the action on the detected structure.

U.S. 5,946,647 (Data Detector)

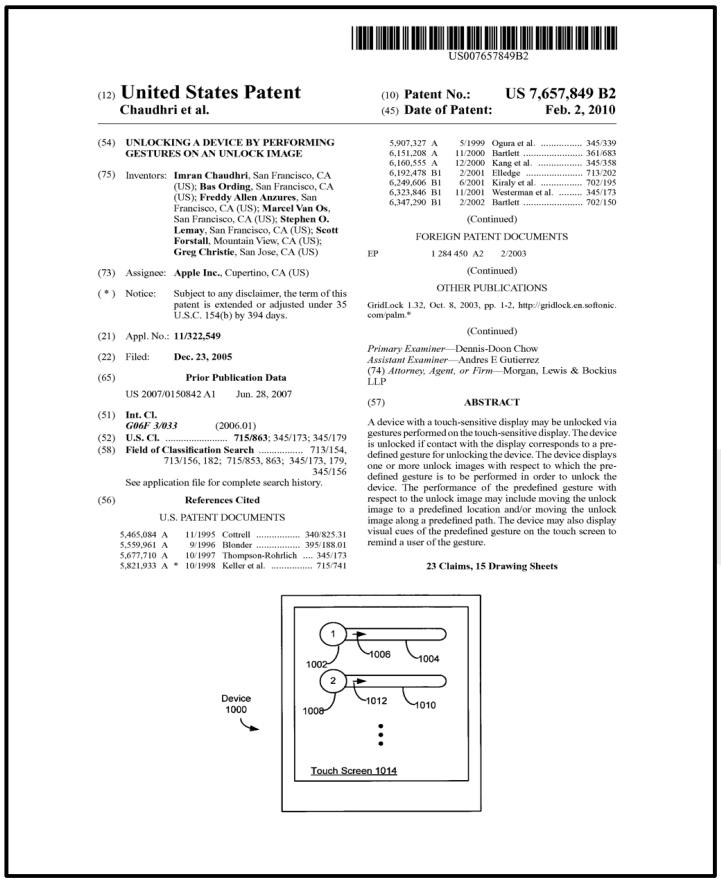
‘647 describes detecting phone numbers, e-mail addresses, and URLs in plain text messages and launching the appropriate program





U.S. 7,657,849 (Unlock Gesture)

- "Unlocking A Device By Performing Gestures On An Unlock Image"
- Filing Date: December 23, 2005



US007657849B2

(12) **United States Patent**
Chaudhri et al.

(10) **Patent No.: US 7,657,849 B2**
(45) **Date of Patent: Feb. 2, 2010**

(54) **UNLOCKING A DEVICE BY PERFORMING GESTURES ON AN UNLOCK IMAGE**

(75) Inventors: **Imran Chaudhri**, San Francisco, CA (US); **Bas Ording**, San Francisco, CA (US); **Freddy Allen Anzures**, San Francisco, CA (US); **Marcel Van Os**, San Francisco, CA (US); **Stephen O. Lemay**, San Francisco, CA (US); **Scott Forstall**, Mountain View, CA (US); **Greg Christie**, San Jose, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 394 days.

(21) Appl. No.: **11/322,549**

(22) Filed: **Dec. 23, 2005**

(65) **Prior Publication Data**

US 2007/0150842 A1 Jun. 28, 2007

(51) **Int. Cl.**
G06F 3/033 (2006.01)

(52) **U.S. Cl.** **715/863**; 345/173; 345/179

(58) **Field of Classification Search** 713/154, 713/156, 182; 715/853, 863; 345/173, 179, 345/156

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,465,084	A	11/1995	Cottrell	340/825.31
5,559,961	A	9/1996	Blonder	395/188.01
5,677,710	A	10/1997	Thompson-Rohrlich	345/173
5,821,933	A *	10/1998	Keller et al.	715/741
5,907,327	A	5/1999	Ogura et al.	345/339
6,151,208	A	11/2000	Bartlett	361/683
6,160,555	A	12/2000	Kang et al.	345/358
6,192,478	B1	2/2001	Elledge	713/202
6,249,606	B1	6/2001	Kiraly et al.	702/195
6,323,846	B1	11/2001	Westerman et al.	345/173
6,347,290	B1	2/2002	Bartlett	702/150

(Continued)

FOREIGN PATENT DOCUMENTS

EP 1 284 450 A2 2/2003

(Continued)

OTHER PUBLICATIONS

GridLock 1.32, Oct. 8, 2003, pp. 1-2, http://gridlock.en.softonic.com/palm.*

(Continued)

Primary Examiner—Dennis-Doon Chow
Assistant Examiner—Andres E Gutierrez
(74) *Attorney, Agent, or Firm*—Morgan, Lewis & Bockius LLP

(57) **ABSTRACT**

A device with a touch-sensitive display may be unlocked via gestures performed on the touch-sensitive display. The device is unlocked if contact with the display corresponds to a predefined gesture for unlocking the device. The device displays one or more unlock images with respect to which the predefined gesture is to be performed in order to unlock the device. The performance of the predefined gesture with respect to the unlock image may include moving the unlock image to a predefined location and/or moving the unlock image along a predefined path. The device may also display visual cues of the predefined gesture on the touch screen to remind a user of the gesture.

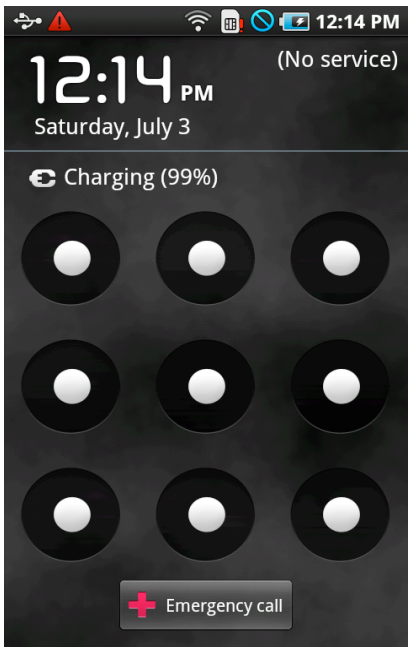
23 Claims, 15 Drawing Sheets

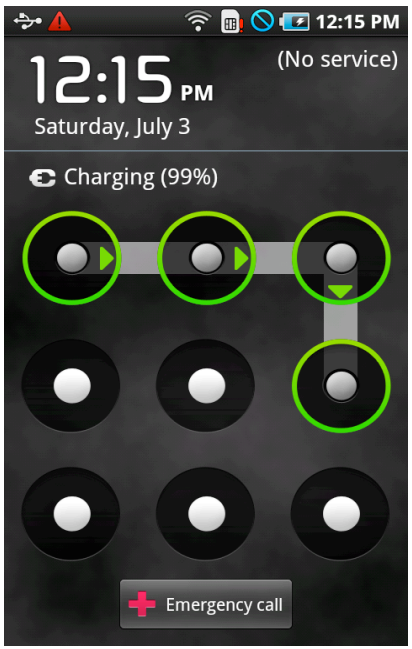
(57) **ABSTRACT**

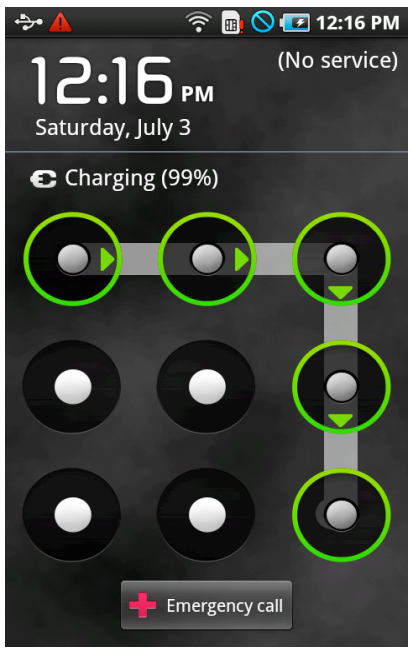
A device with a touch-sensitive display may be unlocked via gestures performed on the touch-sensitive display. The device is unlocked if contact with the display corresponds to a predefined gesture for unlocking the device. The device displays one or more unlock images with respect to which the predefined gesture is to be performed in order to unlock the device. The performance of the predefined gesture with respect to the unlock image may include moving the unlock image to a predefined location and/or moving the unlock image along a predefined path. The device may also display visual cues of the predefined gesture on the touch screen to remind a user of the gesture.

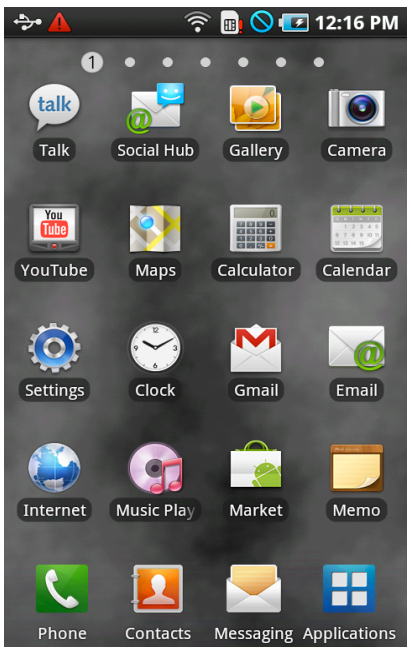
U.S. 7,657,849 (Unlock Gesture)

‘849 patent involves using a slide to unlock gesture on a touch-sensitive display to unlock the phone for use

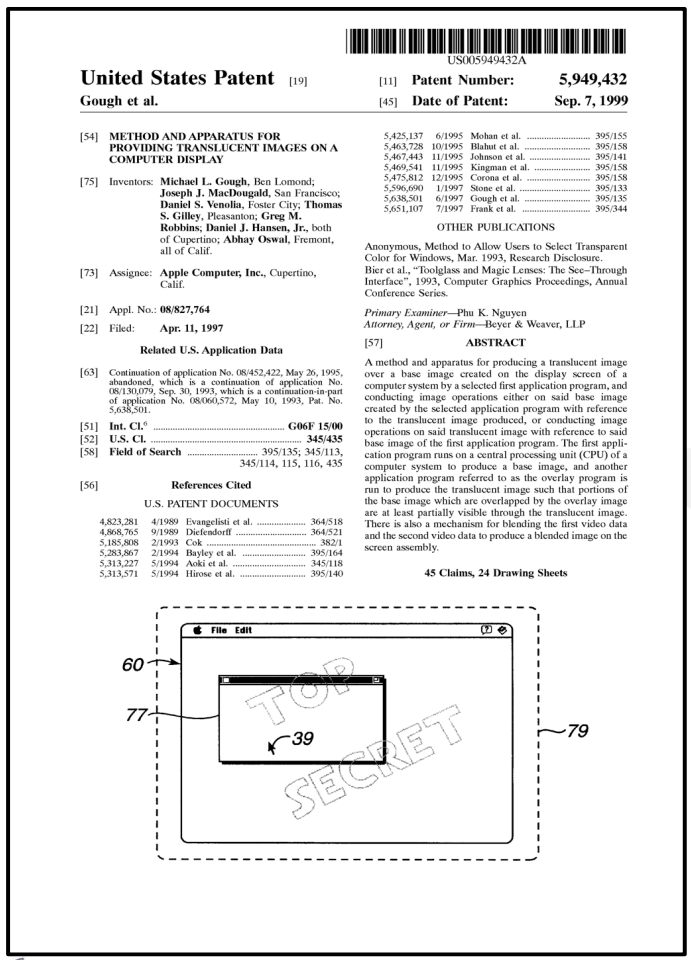








U.S. 5,949,432 & U.S. 6,072,489 (Translucent GUI)



US005949432A

United States Patent [19]
Gough et al.

[11] **Patent Number: 5,949,432**
[45] **Date of Patent: Sep. 7, 1999**

[54] **METHOD AND APPARATUS FOR PROVIDING TRANSLUCENT IMAGES ON A COMPUTER DISPLAY**

[75] Inventors: **Michael L. Gough**, Ben Lomond; **Joseph J. MacDougald**, San Francisco; **Daniel S. Venolia**, Foster City; **Thomas S. Gilley**, Pleasanton; **Greg M. Robbins**; **Daniel J. Hansen, Jr.**, both of Cupertino; **Abhay Oswal**, Fremont, all of Calif.

[73] Assignee: **Apple Computer, Inc.**, Cupertino, Calif.

[21] Appl. No.: **08/827,764**

[22] Filed: **Apr. 11, 1997**

Related U.S. Application Data

[63] Continuation of application No. 08/452,422, May 26, 1995, abandoned, which is a continuation of application No. 08/130,079, Sep. 30, 1993, which is a continuation-in-part of application No. 08/060,572, May 10, 1993, Pat. No. 5,638,501.

[51] **Int. Cl.**[6] **G06F 15/00**
[52] **U.S. Cl.** **345/435**
[58] **Field of Search** 395/135; 345/113, 345/114, 115, 116, 435

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,823,281	4/1989	Evangelisti et al.	364/518
4,868,765	9/1989	Diefendorff	364/521
5,185,808	2/1993	Cok	382/1
5,283,867	2/1994	Bayley et al.	395/164
5,313,227	5/1994	Aoki et al.	345/118
5,313,571	5/1994	Hirose et al.	395/140
5,425,137	6/1995	Mohan et al.	395/155
5,463,728	10/1995	Blahut et al.	395/158
5,467,443	11/1995	Johnson et al.	395/141
5,469,541	11/1995	Kingman et al.	395/158
5,475,812	12/1995	Corona et al.	395/158
5,596,690	1/1997	Stone et al.	395/133
5,638,501	6/1997	Gough et al.	395/135
5,651,107	7/1997	Frank et al.	395/344

OTHER PUBLICATIONS

Anonymous, Method to Allow Users to Select Transparent Color for Windows, Mar. 1993, Research Disclosure.
Bier et al., "Toolglass and Magic Lenses: The See–Through Interface", 1993, Computer Graphics Proceedings, Annual Conference Series.

Primary Examiner—Phu K. Nguyen
Attorney, Agent, or Firm—Beyer & Weaver, LLP

[57] **ABSTRACT**

A method and apparatus for producing a translucent image over a base image created on the display screen of a computer system by a selected first application program, and conducting image operations either on said base image created by the selected application program with reference to the translucent image produced, or conducting image operations on said translucent image with reference to said base image of the first application program. The first application program runs on a central processing unit (CPU) of a computer system to produce a base image, and another application program referred to as the overlay program is run to produce the translucent image such that portions of the base image which are overlapped by the overlay image are at least partially visible through the translucent image. There is also a mechanism for blending the first video data and the second video data to produce a blended image on the screen assembly.

45 Claims, 24 Drawing Sheets

- "Method And Apparatus For Providing Translucent Images On A Computer Display"
- Filing Date: April 11, 1997
  - Priority Date: May 10, 1993

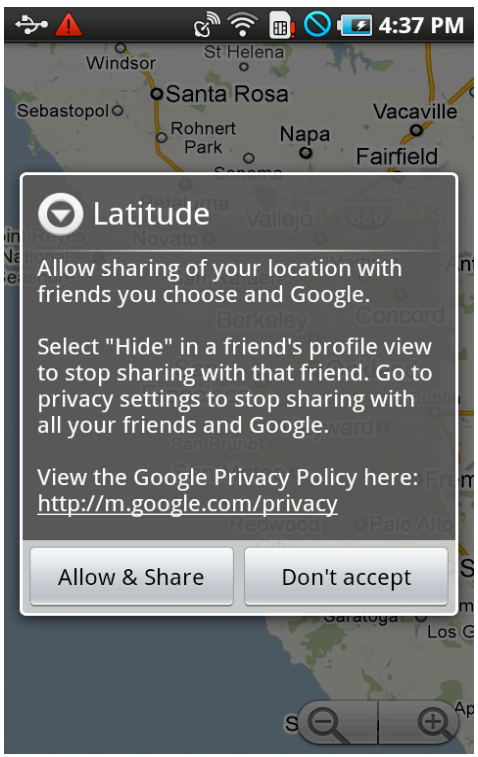
[57] **ABSTRACT**

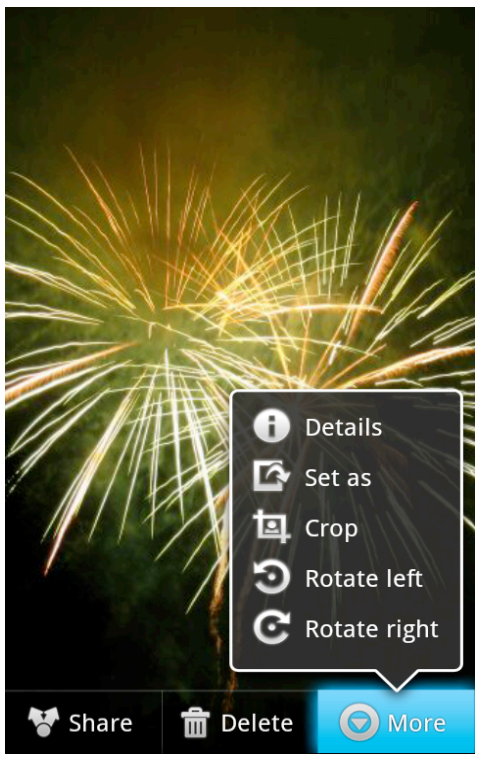
A method and apparatus for producing a translucent image over a base image created on the display screen of a computer system by a selected first application program, and conducting image operations either on said base image created by the selected application program with reference to the translucent image produced, or conducting image operations on said translucent image with reference to said base image of the first application program. The first application program runs on a central processing unit (CPU) of a computer system to produce a base image, and another application program referred to as the overlay program is run to produce the translucent image such that portions of the base image which are overlapped by the overlay image are at least partially visible through the translucent image. There is also a mechanism for blending the first video data and the second video data to produce a blended image on the screen assembly.

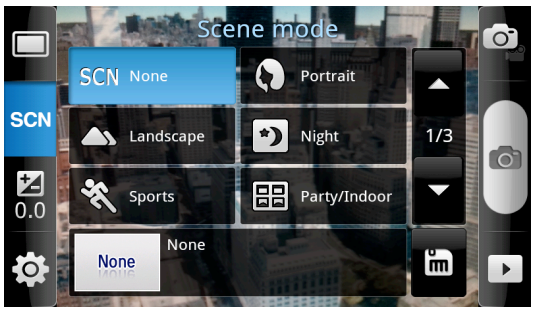
U.S. 5,949,432 & U.S. 6,072,489 (Translucent GUI)

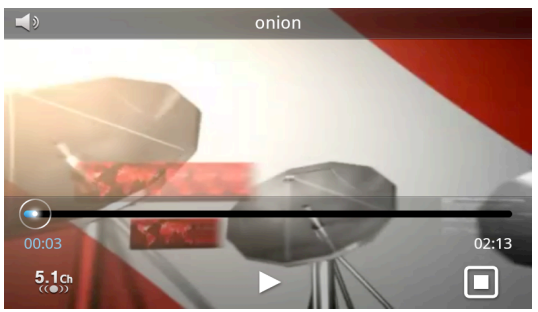
‘432 and ‘489 patents involve producing a translucent image over a base image created on a display screen of a computer system

Samsung Galaxy S Examples

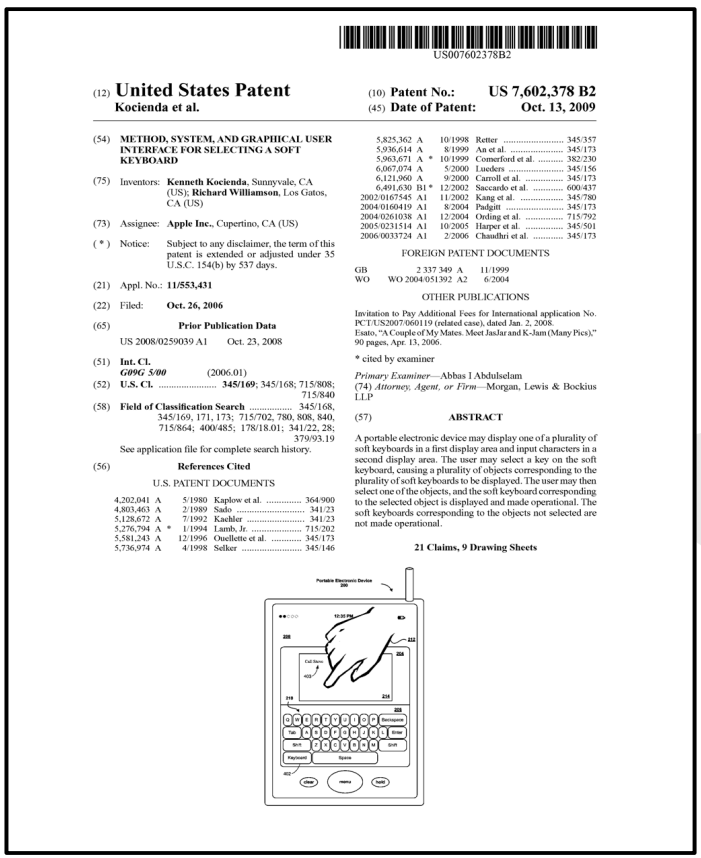








U.S. 7,602,378 (Switchable Soft Keyboard)



US007602378B2

(12) **United States Patent**
Kocienda et al.

(10) **Patent No.: US 7,602,378 B2**
(45) **Date of Patent: Oct. 13, 2009**

(54) **METHOD, SYSTEM, AND GRAPHICAL USER INTERFACE FOR SELECTING A SOFT KEYBOARD**

(75) Inventors: **Kenneth Kocienda**, Sunnyvale, CA (US); **Richard Williamson**, Los Gatos, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 537 days.

(21) Appl. No.: **11/553,431**

(22) Filed: **Oct. 26, 2006**

(65) **Prior Publication Data**

US 2008/0259039 A1 Oct. 23, 2008

(51) **Int. Cl.**
G09G 5/00 (2006.01)

(52) **U.S. Cl.** **345/169**; 345/168; 715/808; 715/840

(58) **Field of Classification Search** 345/168, 345/169, 171, 173; 715/702, 780, 808, 840, 715/864; 400/485; 178/18.01; 341/22, 28; 379/93.19

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,202,041	A	5/1980	Kaplow et al.	364/900
4,803,463	A	2/1989	Sado	341/23
5,128,672	A	7/1992	Kaehler	341/23
5,276,794	A *	1/1994	Lamb, Jr.	715/202
5,581,243	A	12/1996	Ouellette et al.	345/173
5,736,974	A	4/1998	Selker	345/146
5,825,362	A	10/1998	Retter	345/357
5,936,614	A	8/1999	An et al.	345/173
5,963,671	A *	10/1999	Comerford et al.	382/230
6,067,074	A	5/2000	Lueders	345/156
6,121,960	A	9/2000	Carroll et al.	345/173
6,491,630	B1 *	12/2002	Saccardo et al.	600/437
2002/0167545	A1	11/2002	Kang et al.	345/780
2004/0160419	A1	8/2004	Padgitt	345/173
2004/0261038	A1	12/2004	Ording et al.	715/792
2005/0231514	A1	10/2005	Harper et al.	345/501
2006/0033724	A1	2/2006	Chaudhri et al.	345/173

FOREIGN PATENT DOCUMENTS

GB	2 337 349	A	11/1999
WO	WO 2004/051392	A2	6/2004

OTHER PUBLICATIONS

Invitation to Pay Additional Fees for International application No. PCT/US2007/060119 (related case), dated Jan. 2, 2008.
Esato, "A Couple of My Mates. Meet JasJar and K-Jam (Many Pics)," 90 pages, Apr. 13, 2006.

* cited by examiner

Primary Examiner—Abbas I Abdulselam
(74) *Attorney, Agent, or Firm*—Morgan, Lewis & Bockius LLP

(57) **ABSTRACT**

A portable electronic device may display one of a plurality of soft keyboards in a first display area and input characters in a second display area. The user may select a key on the soft keyboard, causing a plurality of objects corresponding to the plurality of soft keyboards to be displayed. The user may then select one of the objects, and the soft keyboard corresponding to the selected object is displayed and made operational. The soft keyboards corresponding to the objects not selected are not made operational.

21 Claims, 9 Drawing Sheets

- "Method, System, And Graphical User Interface For Selecting Soft Keyboard"
- Filing Date: December 26, 2006

(57) **ABSTRACT**

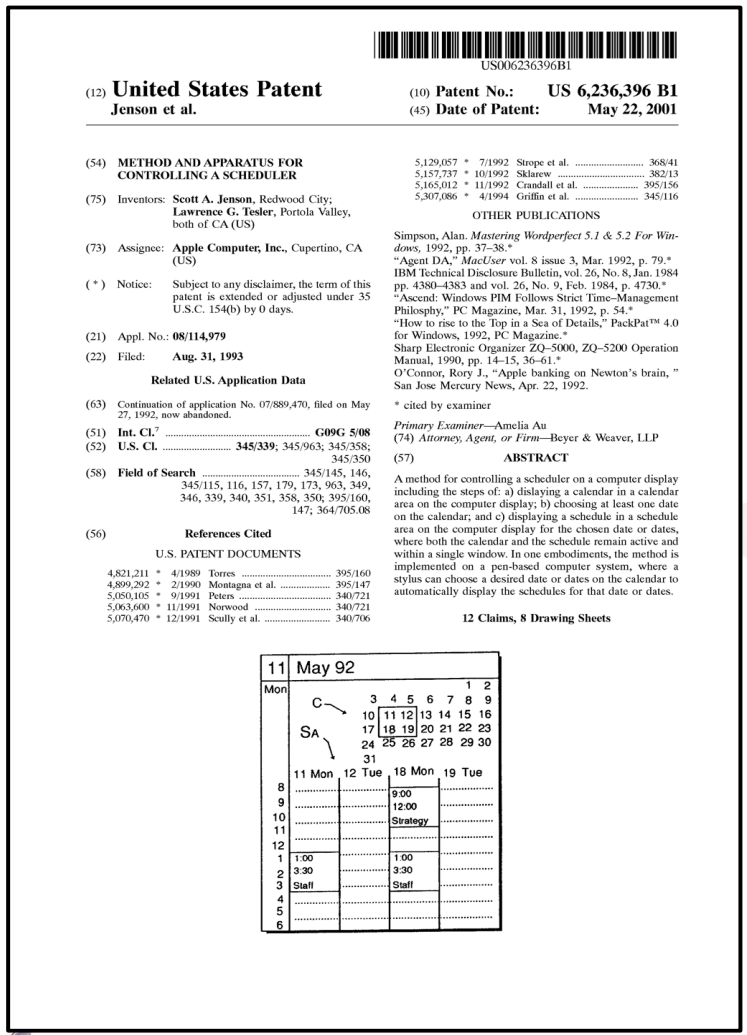
A portable electronic device may display one of a plurality of soft keyboards in a first display area and input characters in a second display area. The user may select a key on the soft keyboard, causing a plurality of objects corresponding to the plurality of soft keyboards to be displayed. The user may then select one of the objects, and the soft keyboard corresponding to the selected object is displayed and made operational. The soft keyboards corresponding to the objects not selected are not made operational.

U.S. 7,602,378 (Switchable Soft Keyboard)

‘378 patent describes a plurality of soft keyboards selected by a plurality of simultaneously displayed objects

Samsung Galaxy S has several infringing applications including contacts and e-mail

U.S. 6,236,396 (Calendar Scheduler)



US006236396B1

(12) **United States Patent**
Jenson et al.

(10) **Patent No.: US 6,236,396 B1**
(45) **Date of Patent: May 22, 2001**

(54) **METHOD AND APPARATUS FOR CONTROLLING A SCHEDULER**

(75) Inventors: **Scott A. Jenson**, Redwood City; **Lawrence G. Tesler**, Portola Valley, both of CA (US)

(73) Assignee: **Apple Computer, Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **08/114,979**

(22) Filed: **Aug. 31, 1993**

Related U.S. Application Data

(63) Continuation of application No. 07/889,470, filed on May 27, 1992, now abandoned.

(51) **Int. Cl.**[7] **G09G 5/08**

(52) **U.S. Cl.** **345/339**; 345/963; 345/358; 345/350

(58) **Field of Search** 345/145, 146, 345/115, 116, 157, 179, 173, 963, 349, 346, 339, 340, 351, 358, 350; 395/160, 147; 364/705.08

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,821,211	*	4/1989	Torres	395/160
4,899,292	*	2/1990	Montagna et al.	395/147
5,050,105	*	9/1991	Peters	340/721
5,063,600	*	11/1991	Norwood	340/721
5,070,470	*	12/1991	Scully et al.	340/706
5,129,057	*	7/1992	Strope et al.	368/41
5,157,737	*	10/1992	Sklarew	382/13
5,165,012	*	11/1992	Crandall et al.	395/156
5,307,086	*	4/1994	Griffin et al.	345/116

OTHER PUBLICATIONS

Simpson, Alan. *Mastering Wordperfect 5.1 & 5.2 For Windows*, 1992, pp. 37–38.*
"Agent DA," *MacUser* vol. 8 issue 3, Mar. 1992, p. 79.*
IBM Technical Disclosure Bulletin, vol. 26, No. 8, Jan. 1984 pp. 4380–4383 and vol. 26, No. 9, Feb. 1984, p. 4730.*
"Ascend: Windows PIM Follows Strict Time–Management Philosphy," PC Magazine, Mar. 31, 1992, p. 54.*
"How to rise to the Top in a Sea of Details," PackPat™ 4.0 for Windows, 1992, PC Magazine.*
Sharp Electronic Organizer ZQ–5000, ZQ–5200 Operation Manual, 1990, pp. 14–15, 36–61.*
O'Connor, Rory J., "Apple banking on Newton's brain," San Jose Mercury News, Apr. 22, 1992.

* cited by examiner

Primary Examiner—Amelia Au
(74) *Attorney, Agent, or Firm*—Beyer & Weaver, LLP

(57) **ABSTRACT**

A method for controlling a scheduler on a computer display including the steps of: a) dislaying a calendar in a calendar area on the computer display; b) choosing at least one date on the calendar; and c) displaying a schedule in a schedule area on the computer display for the chosen date or dates, where both the calendar and the schedule remain active and within a single window. In one embodiments, the method is implemented on a pen-based computer system, where a stylus can choose a desired date or dates on the calendar to automatically display the schedules for that date or dates.

12 Claims, 8 Drawing Sheets

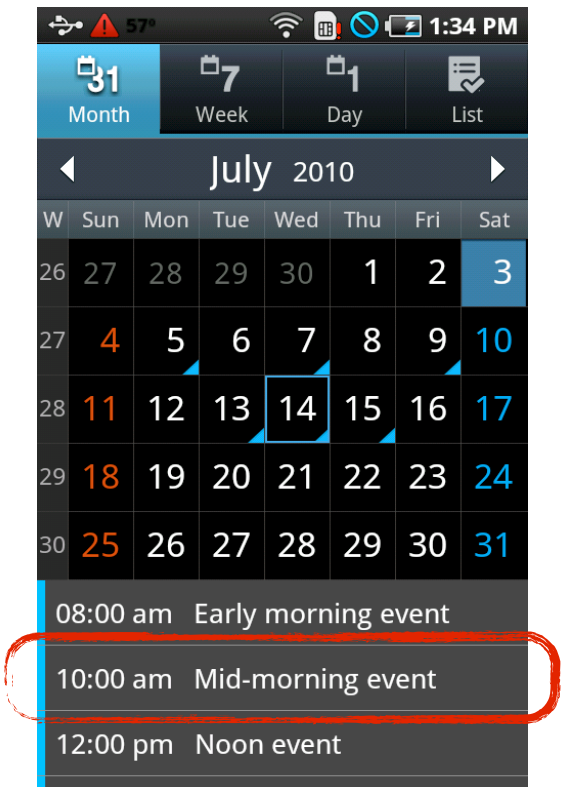
- "Method and Apparatus for Controlling a Scheduler"
- Filing Date: August 31, 1993
  - Priority Date: May 27, 1992

(57) **ABSTRACT**

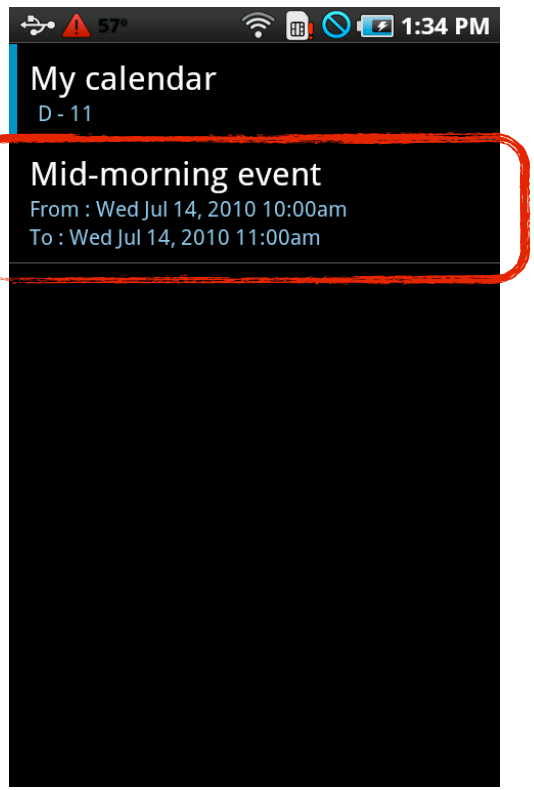
A method for controlling a scheduler on a computer display including the steps of: a) dislaying a calendar in a calendar area on the computer display; b) choosing at least one date on the calendar; and c) displaying a schedule in a schedule area on the computer display for the chosen date or dates, where both the calendar and the schedule remain active and within a single window. In one embodiments, the method is implemented on a pen-based computer system, where a stylus can choose a desired date or dates on the calendar to automatically display the schedules for that date or dates.

U.S. 6,236,396 (Calendar Scheduler)

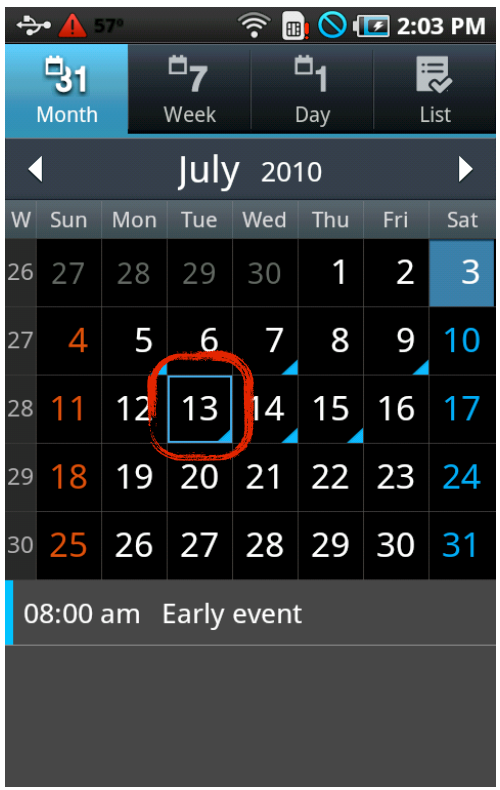
‘396 patent describes entering information into a calendar scheduler for a specific date while still being able to view the calendar



Samsung Galaxy S
Month and List View

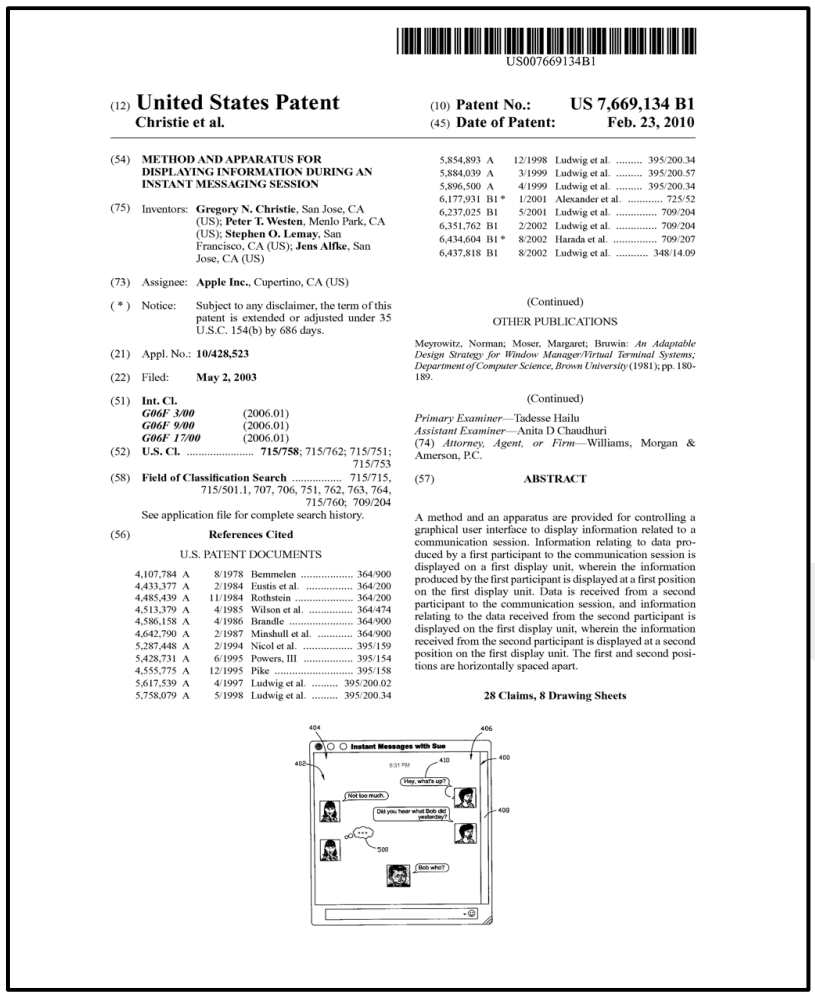


Samsung Galaxy S
List View is Active



Samsung Galaxy S
Month View is Active

U.S. 7,669,134 (Messaging UI)



US007669134B1

(12) **United States Patent**
Christie et al.

(10) **Patent No.: US 7,669,134 B1**
(45) **Date of Patent: Feb. 23, 2010**

(54) **METHOD AND APPARATUS FOR DISPLAYING INFORMATION DURING AN INSTANT MESSAGING SESSION**

(75) Inventors: **Gregory N. Christie**, San Jose, CA (US); **Peter T. Westen**, Menlo Park, CA (US); **Stephen O. Lemay**, San Francisco, CA (US); **Jens Alfke**, San Jose, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 686 days.

(21) Appl. No.: **10/428,523**

(22) Filed: **May 2, 2003**

(51) **Int. Cl.**
G06F 3/00 (2006.01)
G06F 9/00 (2006.01)
G06F 17/00 (2006.01)

(52) **U.S. Cl.** **715/758**; 715/762; 715/751; 715/753

(58) **Field of Classification Search** 715/715, 715/501.1, 707, 706, 751, 762, 763, 764, 715/760; 709/204
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,107,784	A	8/1978	Bemmelen	364/900
4,433,377	A	2/1984	Eustis et al.	364/200
4,485,439	A	11/1984	Rothstein	364/200
4,513,379	A	4/1985	Wilson et al.	364/474
4,586,158	A	4/1986	Brandle	364/900
4,642,790	A	2/1987	Minshull et al.	364/900
5,287,448	A	2/1994	Nicol et al.	395/159
5,428,731	A	6/1995	Powers, III	395/154
4,555,775	A	12/1995	Pike	395/158
5,617,539	A	4/1997	Ludwig et al.	395/200.02
5,758,079	A	5/1998	Ludwig et al.	395/200.34
5,854,893	A	12/1998	Ludwig et al.	395/200.34
5,884,039	A	3/1999	Ludwig et al.	395/200.57
5,896,500	A	4/1999	Ludwig et al.	395/200.34
6,177,931	B1 *	1/2001	Alexander et al.	725/52
6,237,025	B1	5/2001	Ludwig et al.	709/204
6,351,762	B1	2/2002	Ludwig et al.	709/204
6,434,604	B1 *	8/2002	Harada et al.	709/207
6,437,818	B1	8/2002	Ludwig et al.	348/14.09

(Continued)

OTHER PUBLICATIONS

Meyrowitz, Norman; Moser, Margaret; Bruwin: *An Adaptable Design Strategy for Window Manager/Virtual Terminal Systems; Department of Computer Science, Brown University* (1981); pp. 180-189.

(Continued)

Primary Examiner—Tadesse Hailu
Assistant Examiner—Anita D Chaudhuri
(74) *Attorney, Agent, or Firm*—Williams, Morgan & Amerson, P.C.

(57) **ABSTRACT**

A method and an apparatus are provided for controlling a graphical user interface to display information related to a communication session. Information relating to data produced by a first participant to the communication session is displayed on a first display unit, wherein the information produced by the first participant is displayed at a first position on the first display unit. Data is received from a second participant to the communication session, and information relating to the data received from the second participant is displayed on the first display unit, wherein the information received from the second participant is displayed at a second position on the first display unit. The first and second positions are horizontally spaced apart.

28 Claims, 8 Drawing Sheets

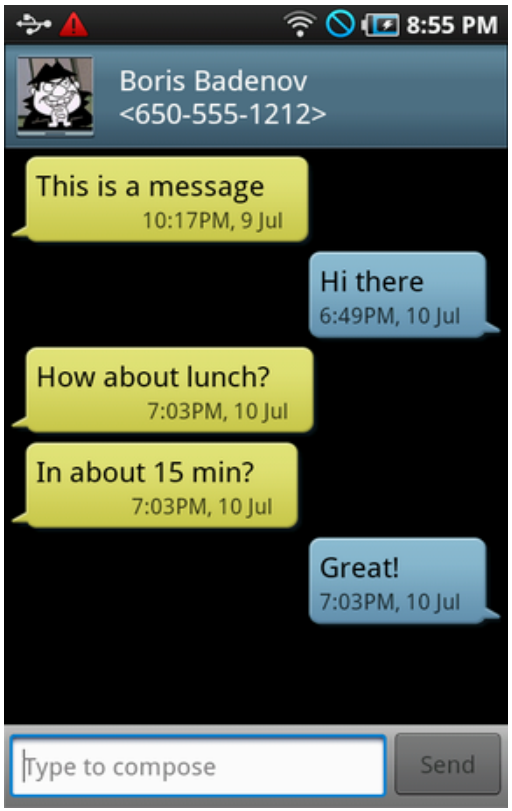
- "Method And apparatus For Displaying Information During An Instant Messaging Session"
- Filing Date: May 2, 2003

(57) **ABSTRACT**

A method and an apparatus are provided for controlling a graphical user interface to display information related to a communication session. Information relating to data produced by a first participant to the communication session is displayed on a first display unit, wherein the information produced by the first participant is displayed at a first position on the first display unit. Data is received from a second participant to the communication session, and information relating to the data received from the second participant is displayed on the first display unit, wherein the information received from the second participant is displayed at a second position on the first display unit. The first and second positions are horizontally spaced apart.

U.S. 7,669,134 (Messaging UI)

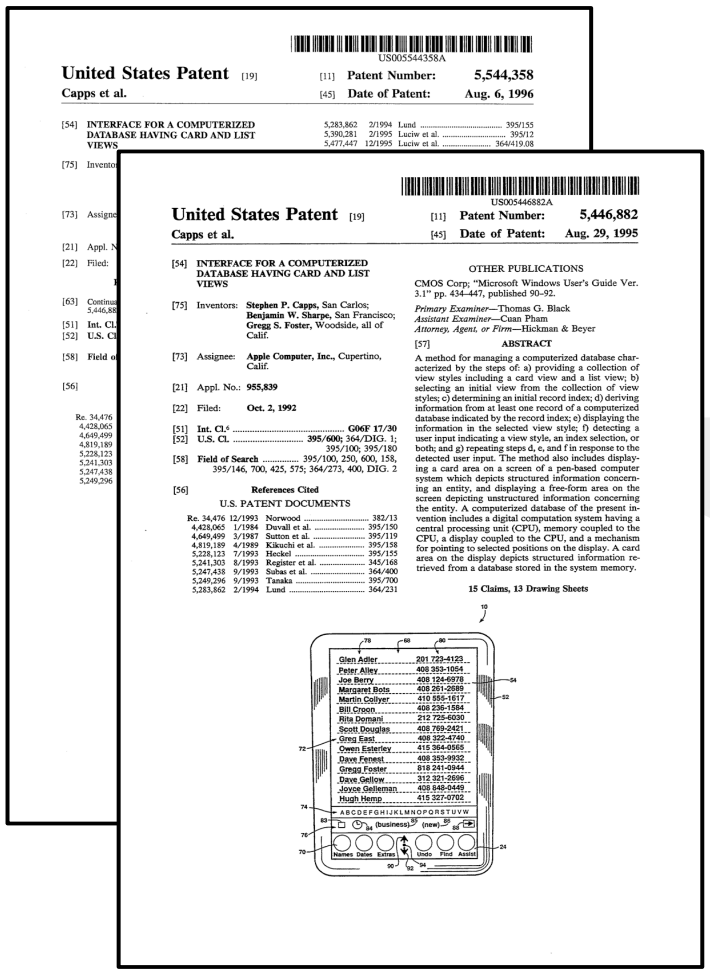
‘134 patent describes messages from the two participants grouped in two columns and arranged temporally down the page



Samsung Galaxy S

Apple iPhone

U.S. 5,544,358 & U.S. 5,446,882 (Address Book UI)



United States Patent [19]
Capps et al.

[11] Patent Number: 5,544,358
[45] Date of Patent: Aug. 6, 1996

[54] INTERFACE FOR A COMPUTERIZED DATABASE HAVING CARD AND LIST VIEWS

United States Patent [19]
Capps et al.

[11] Patent Number: 5,446,882
[45] Date of Patent: Aug. 29, 1995

[54] INTERFACE FOR A COMPUTERIZED DATABASE HAVING CARD AND LIST VIEWS

[75] Inventors: Stephen P. Capps, San Carlos; Benjamin W. Sharpe, San Francisco; Gregg S. Foster, Woodside, all of Calif.

[73] Assignee: Apple Computer, Inc., Cupertino, Calif.

[21] Appl. No.: 955,839

[22] Filed: Oct. 2, 1992

[51] Int. Cl.[5] G06F 17/30
[52] U.S. Cl. 395/600; 364/DIG. 1; 395/100; 395/180
[58] Field of Search ..... 395/100, 250, 600, 158, 395/146, 700, 425, 575; 364/273, 400, DIG. 2

[56] References Cited
U.S. PATENT DOCUMENTS

Re. 34,476	12/1993	Norwood	382/13
4,428,065	1/1984	Duvall et al.	395/150
4,649,499	3/1987	Sutton et al.	395/119
4,819,189	4/1989	Kikuchi et al.	395/158
5,228,123	7/1993	Heckel	395/155
5,241,303	8/1993	Register et al.	345/168
5,247,438	9/1993	Subas et al.	364/400
5,249,296	9/1993	Tanaka	395/700
5,283,862	2/1994	Lund	364/231
5,390,281	2/1995	Luciw et al.	395/12
5,477,447	12/1995	Luciw et al.	364/419.08

OTHER PUBLICATIONS

CMOS Corp; "Microsoft Windows User's Guide Ver. 3.1" pp. 434–447, published 90–92.

Primary Examiner—Thomas G. Black
Assistant Examiner—Cuan Pham
Attorney, Agent, or Firm—Hickman & Beyer

[57] ABSTRACT

A method for managing a computerized database characterized by the steps of: a) providing a collection of view styles including a card view and a list view; b) selecting an initial view from the collection of view styles; c) determining an initial record index; d) deriving information from at least one record of a computerized database indicated by the record index; e) displaying the information in the selected view style; f) detecting a user input indicating a view style, an index selection, or both; and g) repeating steps d, e, and f in response to the detected user input. The method also includes displaying a card area on a screen of a pen-based computer system which depicts structured information concerning an entity, and displaying a free-form area on the screen depicting unstructured information concerning the entity. A computerized database of the present invention includes a digital computation system having a central processing unit (CPU), memory coupled to the CPU, a display coupled to the CPU, and a mechanism for pointing to selected positions on the display. A card area on the display depicts structured information retrieved from a database stored in the system memory.

15 Claims, 13 Drawing Sheets

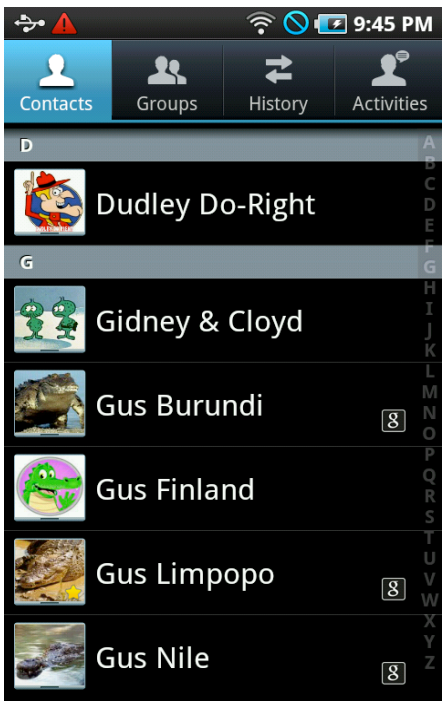
- "Interface For A Computerized Database Having Card And List Views"
- Filing Date: October 2, 1992

[57] **ABSTRACT**

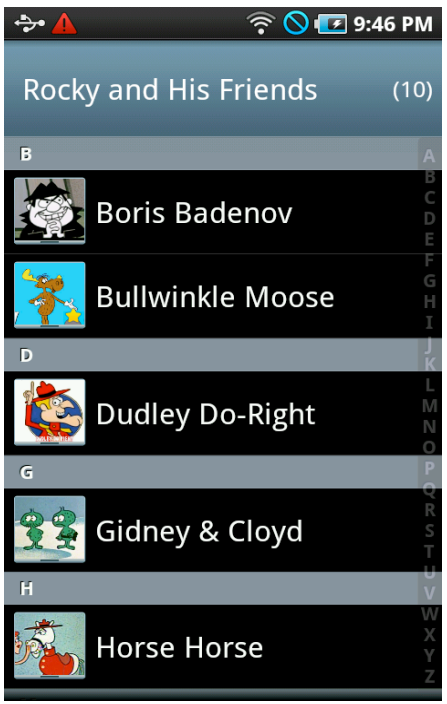
A method for managing a computerized database characterized by the steps of: a) providing a collection of view styles including a card view and a list view; b) selecting an initial view from the collection of view styles; c) determining an initial record index; d) deriving information from at least one record of a computerized database indicated by the record index; e) displaying the information in the selected view style; f) detecting a user input indicating a view style, an index selection, or both; and g) repeating steps d, e, and f in response to the detected user input. The method also includes displaying a card area on a screen of a pen-based computer system which depicts structured information concerning an entity, and displaying a free-form area on the screen depicting unstructured information concerning the entity. A computerized database of the present invention includes a digital computation system having a central processing unit (CPU), memory coupled to the CPU, a display coupled to the CPU, and a mechanism for pointing to selected positions on the display. A card area on the display depicts structured information retrieved from a database stored in the system memory.

U.S. 5,544,358 & U.S. 5,446,882 (Address Book UI)

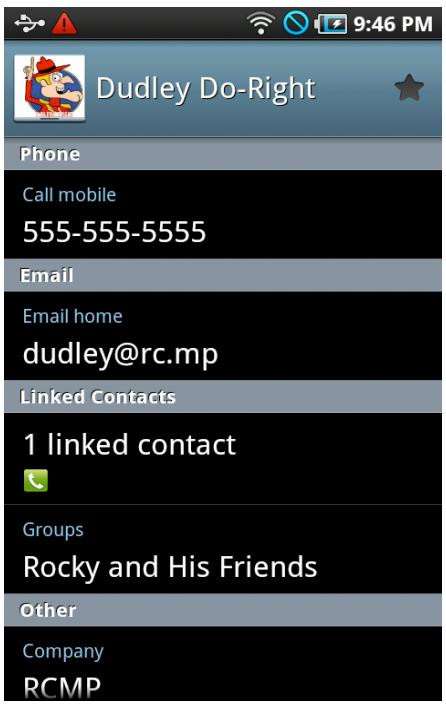
‘358 and ‘882 patents describe an address book with filtering capability and a card view with structured (fields) and unstructured (sketches) information



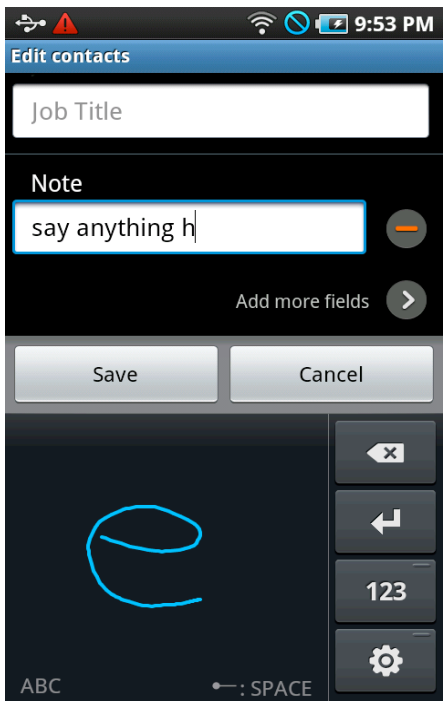
Galaxy S Contacts



Galaxy S Filtered Contacts

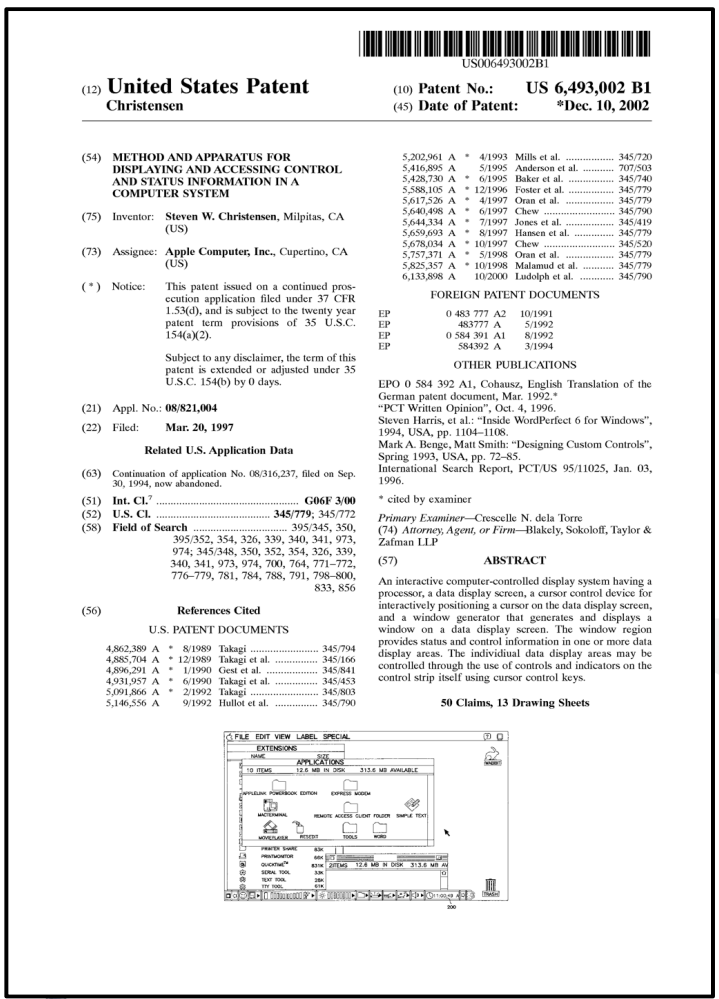


Galaxy S Structured Information



Galaxy S Unstructured Information

U.S. 6,493,002 (Status Bar)



US006493002B1

(12) **United States Patent**
Christensen

(10) **Patent No.: US 6,493,002 B1**
(45) **Date of Patent: *Dec. 10, 2002**

(54) **METHOD AND APPARATUS FOR DISPLAYING AND ACCESSING CONTROL AND STATUS INFORMATION IN A COMPUTER SYSTEM**

(75) Inventor: **Steven W. Christensen**, Milpitas, CA (US)

(73) Assignee: **Apple Computer, Inc.**, Cupertino, CA (US)

(*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **08/821,004**

(22) Filed: **Mar. 20, 1997**

Related U.S. Application Data

(63) Continuation of application No. 08/316,237, filed on Sep. 30, 1994, now abandoned.

(51) **Int. Cl.**[7] **G06F 3/00**

(52) **U.S. Cl.** **345/779**; 345/772

(58) **Field of Search** 395/345, 350, 395/352, 354, 326, 339, 340, 341, 973, 974; 345/348, 350, 352, 354, 326, 339, 340, 341, 973, 974, 700, 764, 771–772, 776–779, 781, 784, 788, 791, 798–800, 833, 856

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,862,389	A *	8/1989	Takagi	345/794
4,885,704	A *	12/1989	Takagi et al.	345/166
4,896,291	A *	1/1990	Gest et al.	345/841
4,931,957	A *	6/1990	Takagi et al.	345/453
5,091,866	A *	2/1992	Takagi	345/803
5,146,556	A	9/1992	Hullot et al.	345/790
5,202,961	A *	4/1993	Mills et al.	345/720
5,416,895	A	5/1995	Anderson et al.	707/503
5,428,730	A *	6/1995	Baker et al.	345/740
5,588,105	A *	12/1996	Foster et al.	345/779
5,617,526	A *	4/1997	Oran et al.	345/779
5,640,498	A *	6/1997	Chew	345/790
5,644,334	A *	7/1997	Jones et al.	345/419
5,659,693	A *	8/1997	Hansen et al.	345/779
5,678,034	A *	10/1997	Chew	345/520
5,757,371	A *	5/1998	Oran et al.	345/779
5,825,357	A *	10/1998	Malamud et al.	345/779
6,133,898	A	10/2000	Ludolph et al.	345/790

FOREIGN PATENT DOCUMENTS

EP	0 483 777 A2	10/1991	
EP	483777 A	5/1992	
EP	0 584 391 A1	8/1992	
EP	584392 A	3/1994	

OTHER PUBLICATIONS

EPO 0 584 392 A1, Cohausz, English Translation of the German patent document, Mar. 1992.*
"PCT Written Opinion", Oct. 4, 1996.
Steven Harris, et al.: "Inside WordPerfect 6 for Windows", 1994, USA, pp. 1104–1108.
Mark A. Benge, Matt Smith: "Designing Custom Controls", Spring 1993, USA, pp. 72–85.
International Search Report, PCT/US 95/11025, Jan. 03, 1996.

* cited by examiner

Primary Examiner—Crescelle N. dela Torre
(74) *Attorney, Agent, or Firm*—Blakely, Sokoloff, Taylor & Zafman LLP

(57) **ABSTRACT**

An interactive computer-controlled display system having a processor, a data display screen, a cursor control device for interactively positioning a cursor on the data display screen, and a window generator that generates and displays a window on a data display screen. The window region provides status and control information in one or more data display areas. The individiual data display areas may be controlled through the use of controls and indicators on the control strip itself using cursor control keys.

50 Claims, 13 Drawing Sheets

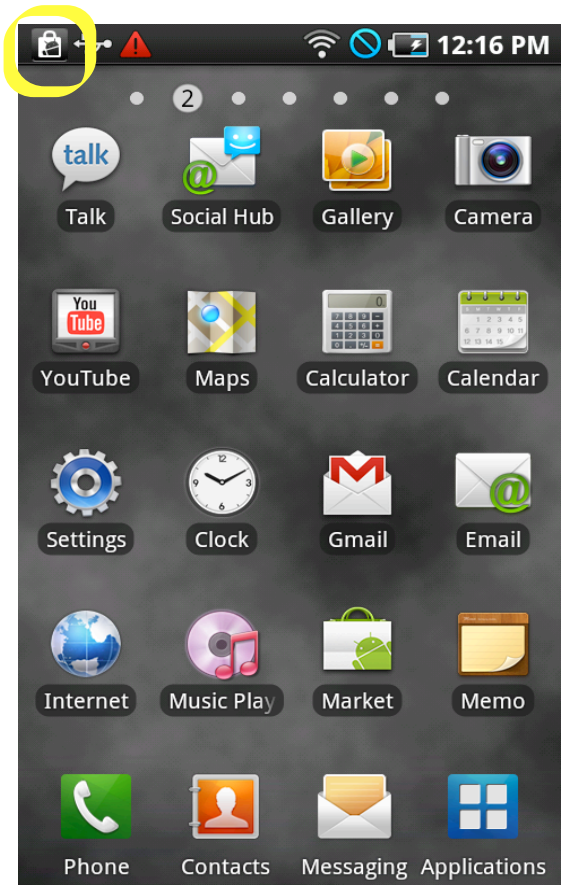
- "Method And Apparatus For Displaying And Accessing Control And Status Information In A Computer System"
- Filing Date: March 20, 1997
  - Priority: September 30, 1994

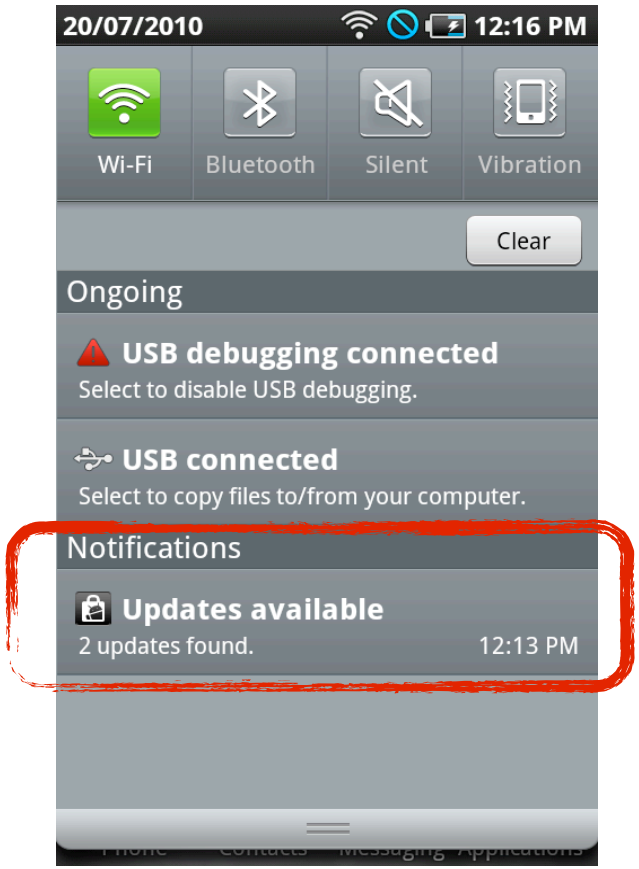
(57) **ABSTRACT**

An interactive computer-controlled display system having a processor, a data display screen, a cursor control device for interactively positioning a cursor on the data display screen, and a window generator that generates and displays a window on a data display screen. The window region provides status and control information in one or more data display areas. The individiual data display areas may be controlled through the use of controls and indicators on the control strip itself using cursor control keys.

U.S. 6,493,002 (Status Bar)

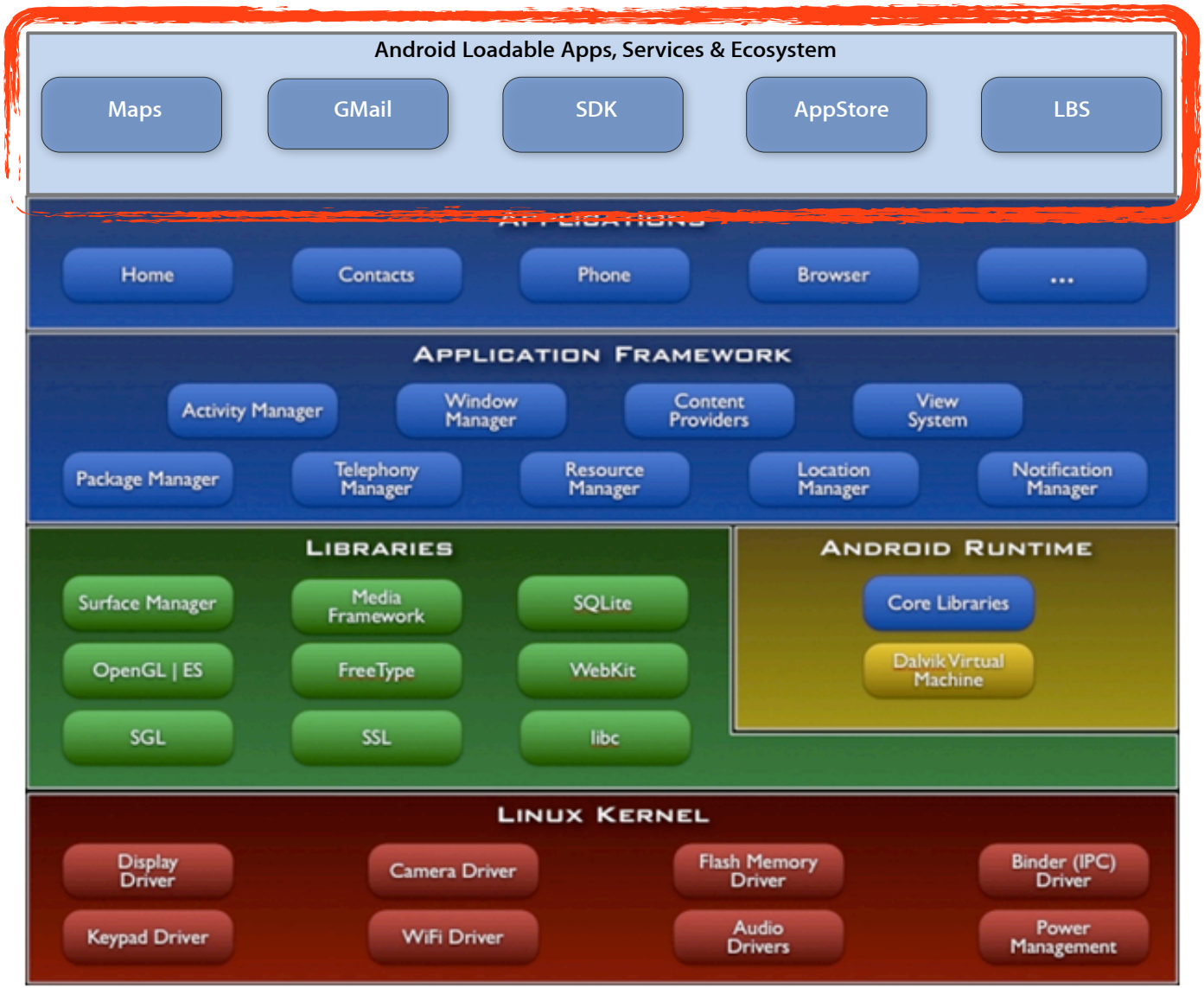
- '002 Patent describes the use of a status and control bar
- Android market application infringes





Notification Window

Android Stack



Highlights of Example Apple Patents Infringed by Android Ecosystem

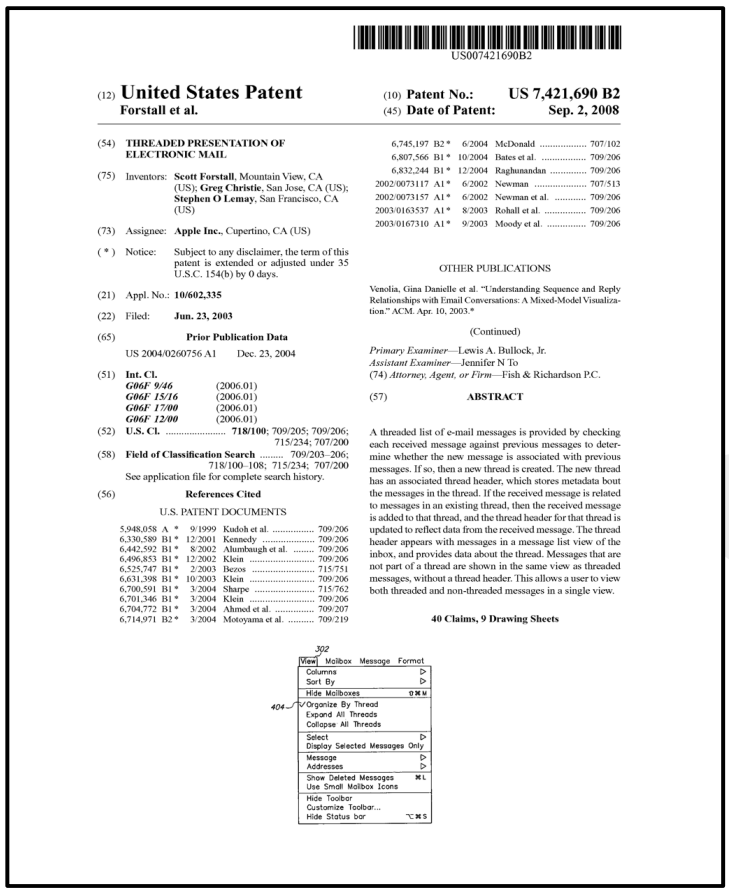
U.S. 7,421,690 (Threaded E-Mail)

U.S. 5,926,190 (Street View Image Synthesis)

U.S. 7,187,997 (Situational Location)

U.S. 7,421,690 (Threaded E-Mail)

- "Threaded Presentation of Electronic Mail"
- Filing Date: June 23, 2003



US007421690B2

(12) **United States Patent**
Forstall et al.

(10) **Patent No.: US 7,421,690 B2**
(45) **Date of Patent: Sep. 2, 2008**

(54) **THREADED PRESENTATION OF ELECTRONIC MAIL**

(75) Inventors: **Scott Forstall**, Mountain View, CA (US); **Greg Christie**, San Jose, CA (US); **Stephen O Lemay**, San Francisco, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **10/602,335**

(22) Filed: **Jun. 23, 2003**

(65) **Prior Publication Data**

US 2004/0260756 A1 Dec. 23, 2004

(51) **Int. Cl.**
G06F 9/46 (2006.01)
G06F 15/16 (2006.01)
G06F 17/00 (2006.01)
G06F 12/00 (2006.01)

(52) **U.S. Cl.** **718/100**; 709/205; 709/206; 715/234; 707/200

(58) **Field of Classification Search** 709/203–206; 718/100–108; 715/234; 707/200
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,948,058	A *	9/1999	Kudoh et al.	709/206
6,330,589	B1 *	12/2001	Kennedy	709/206
6,442,592	B1 *	8/2002	Alumbaugh et al.	709/206
6,496,853	B1 *	12/2002	Klein	709/206
6,525,747	B1 *	2/2003	Bezos	715/751
6,631,398	B1 *	10/2003	Klein	709/206
6,700,591	B1 *	3/2004	Sharpe	715/762
6,701,346	B1 *	3/2004	Klein	709/206
6,704,772	B1 *	3/2004	Ahmed et al.	709/207
6,714,971	B2 *	3/2004	Motoyama et al.	709/219
6,745,197	B2 *	6/2004	McDonald	707/102
6,807,566	B1 *	10/2004	Bates et al.	709/206
6,832,244	B1 *	12/2004	Raghunandan	709/206
2002/0073117	A1 *	6/2002	Newman	707/513
2002/0073157	A1 *	6/2002	Newman et al.	709/206
2003/0163537	A1 *	8/2003	Rohall et al.	709/206
2003/0167310	A1 *	9/2003	Moody et al.	709/206

OTHER PUBLICATIONS

Venolia, Gina Danielle et al. "Understanding Sequence and Reply Relationships with Email Conversations: A Mixed-Model Visualization." ACM. Apr. 10, 2003.*

(Continued)

Primary Examiner—Lewis A. Bullock, Jr.
Assistant Examiner—Jennifer N To
(74) *Attorney, Agent, or Firm*—Fish & Richardson P.C.

(57) **ABSTRACT**

A threaded list of e-mail messages is provided by checking each received message against previous messages to determine whether the new message is associated with previous messages. If so, then a new thread is created. The new thread has an associated thread header, which stores metadata bout the messages in the thread. If the received message is related to messages in an existing thread, then the received message is added to that thread, and the thread header for that thread is updated to reflect data from the received messsage. The thread header appears with messages in a message list view of the inbox, and provides data about the thread. Messages that are not part of a thread are shown in the same view as threaded messages, without a thread header. This allows a user to view both threaded and non-threaded messages in a single view.

40 Claims, 9 Drawing Sheets

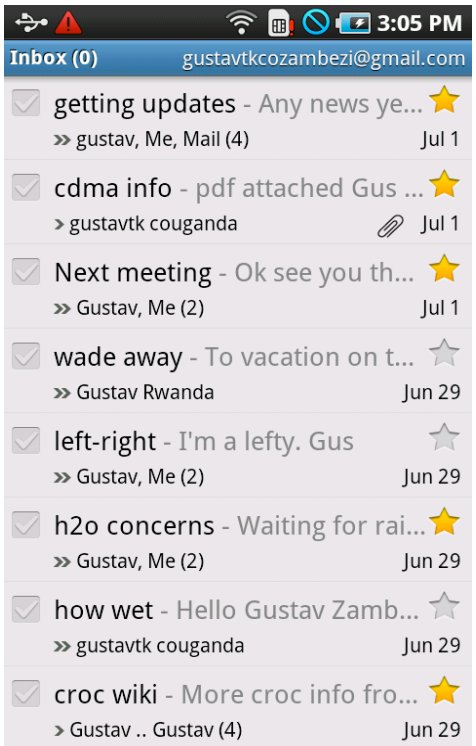
(57) **ABSTRACT**

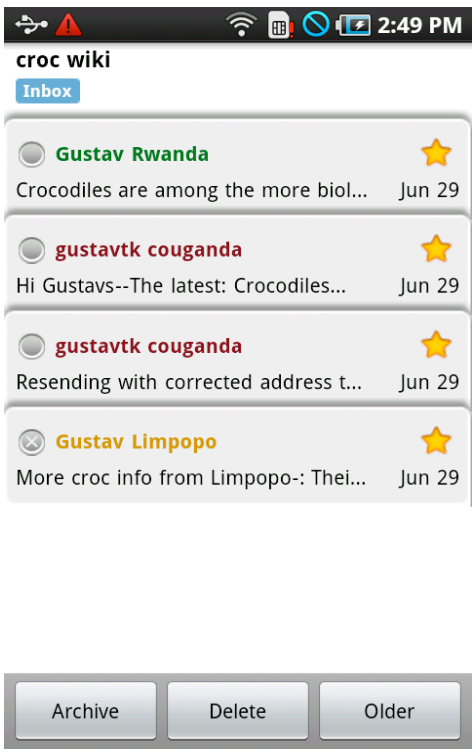
A threaded list of e-mail messages is provided by checking each received message against previous messages to determine whether the new message is associated with previous messages. If so, then a new thread is created. The new thread has an associated thread header, which stores metadata bout the messages in the thread. If the received message is related to messages in an existing thread, then the received message is added to that thread, and the thread header for that thread is updated to reflect data from the received message. The thread header appears with messages in a message list view of the inbox, and provides data about the thread. Messages that are not part of a thread are shown in the same view as threaded messages, without a thread header. This allows a user to view both threaded and non-threaded messages in a single view.

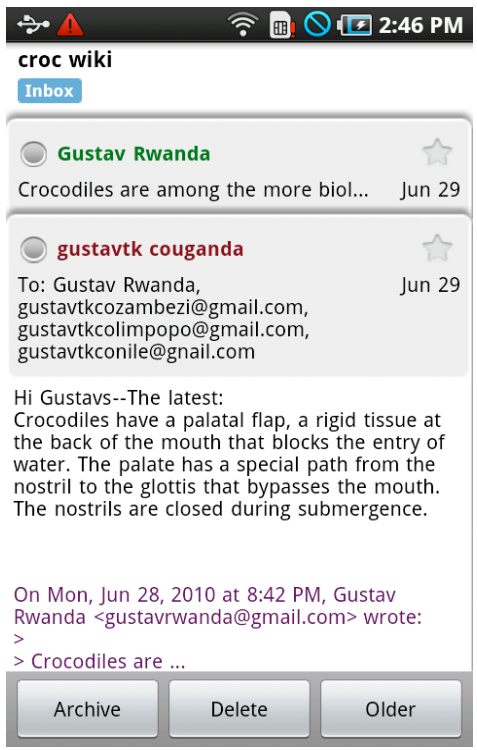
U.S. 7,421,690 (Threaded E-Mail)

‘690 patent relates to a threaded list of e-mail messages that groups together e-mails that are part of the same chain

Samsung Galaxy S Threaded E-mail Examples







U.S. 5,926,190 (Street View Image Synthesis)

US005926190A

United States Patent [19]
Turkowski et al.

[11] **Patent Number: 5,926,190**
[45] **Date of Patent: Jul. 20, 1999**

[54] **METHOD AND SYSTEM FOR SIMULATING MOTION IN A COMPUTER GRAPHICS APPLICATION USING IMAGE REGISTRATION AND VIEW INTERPOLATION**

[75] Inventors: **Kenneth E. Turkowski**, Menlo Park; **Heng-Yeung Shum**, San Jose, both of Calif.

[73] Assignee: **Apple Computer, Inc.**, Cupertino, Calif.

[21] Appl. No.: **08/701,817**

[22] Filed: **Aug. 21, 1996**

[51] **Int. Cl.**[6] **G06F 15/00**
[52] **U.S. Cl.** **345/473**
[58] **Field of Search** 345/435, 440, 345/441, 442, 475, 473

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,818,461 10/1998 Rouet et al. 345/473

OTHER PUBLICATIONS

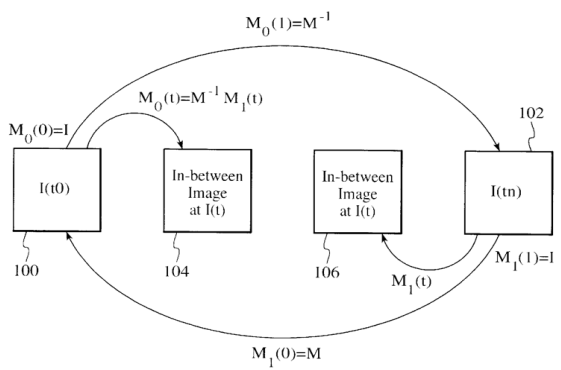
Shenchang Eric Chen; QuickTime VR–An Image–Based Approach to Virtual Environment Navigation; Apple Computer, Inc. Aug. 1995.
Leonard McMillan & GAry Bishop; Plenoptic Modeling: An Image–Based Rendering System; Dept. of Computer Science Univ. of North Carolina Aug. 1995.
Paul S. Heckbert; Basic Texture Mappings and Image Warps; Dept. of Electrical Engineering & Computer Science Univ. of CA Berkeley Mar. 29, 1989.
Richard Szeliski and James Coughlan; Spline–Based Image Registration; Digital Equipment Corp. Cambridge Research Lab Apr., 1994.
Peter Litwinowicz and Gavin Miller; Efficient Techniques for Interactive Texture Placement; Apple Computer, Inc. Jul. 1994.
Thaddeus Beier, Shawn Neely; Feature–Based Image Metamorphosis; Silicon Graphics Computer Systems; Pacific Data Images Jul. 1992.

Primary Examiner—Phu K. Nguyen
Attorney, Agent, or Firm—Sawyer & Associates

[57] **ABSTRACT**

A system and method for simulating motion in an interactive computer application wherein images of a scene are used to render views of the scene from particular viewpoints. The method and system includes registering a first image and a second image using a transformation that overlays common features of both the first and the second image. After the two images are registered, at least one intermediate image is generated by applying a first interpolated transformation to the first image and a second interpolated transformation to the second image. The first image, the intermediate image, and the second image are then sequentially rendered to simulate motion through the scene.

21 Claims, 8 Drawing Sheets



- "Method And System For Simulating Motion In A Computer Graphics Application Using Image Registration And View Interpolation"
- Filing Date: August 21, 1996

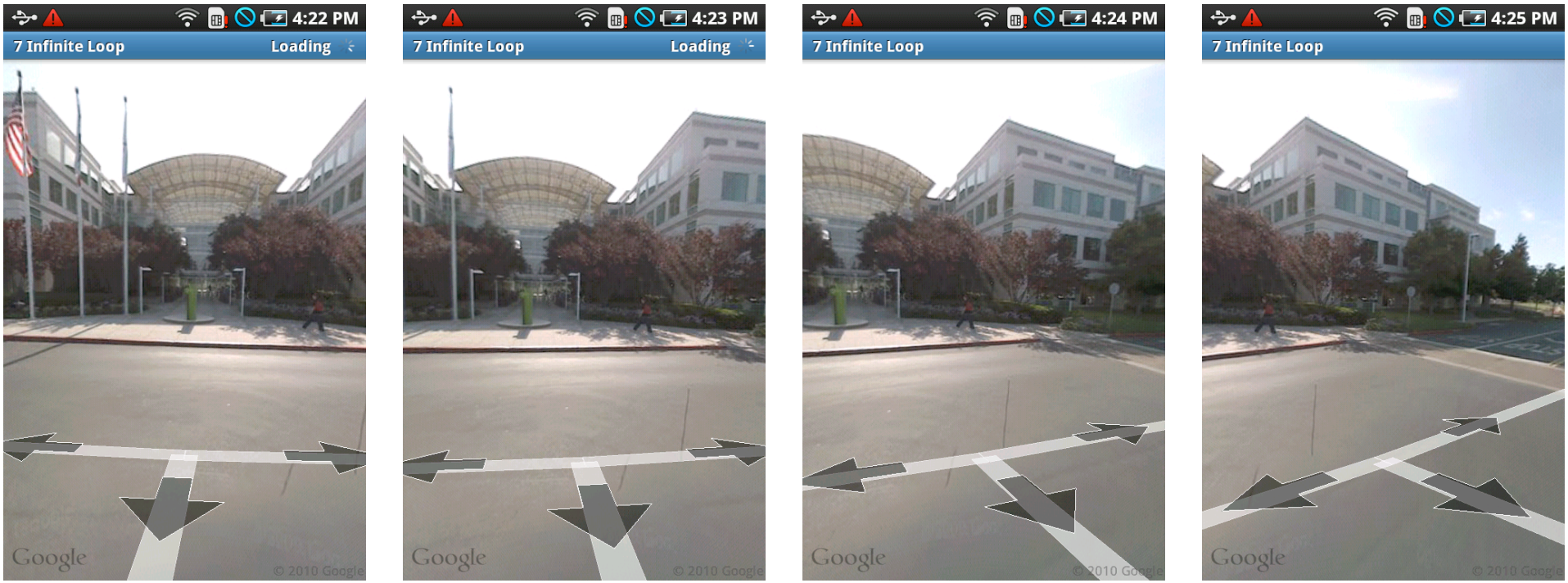
[57] **ABSTRACT**

A system and method for simulating motion in an interactive computer application wherein images of a scene are used to render views of the scene from particular viewpoints. The method and system includes registering a first image and a second image using a transformation that overlays common features of both the first and the second image. After the two images are registered, at least one intermediate image is generated by applying a first interpolated transformation to the first image and a second interpolated transformation to the second image. The first image, the intermediate image, and the second image are then sequentially rendered to simulate motion through the scene.

U.S. 5,926,190 (Street View Image Synthesis)

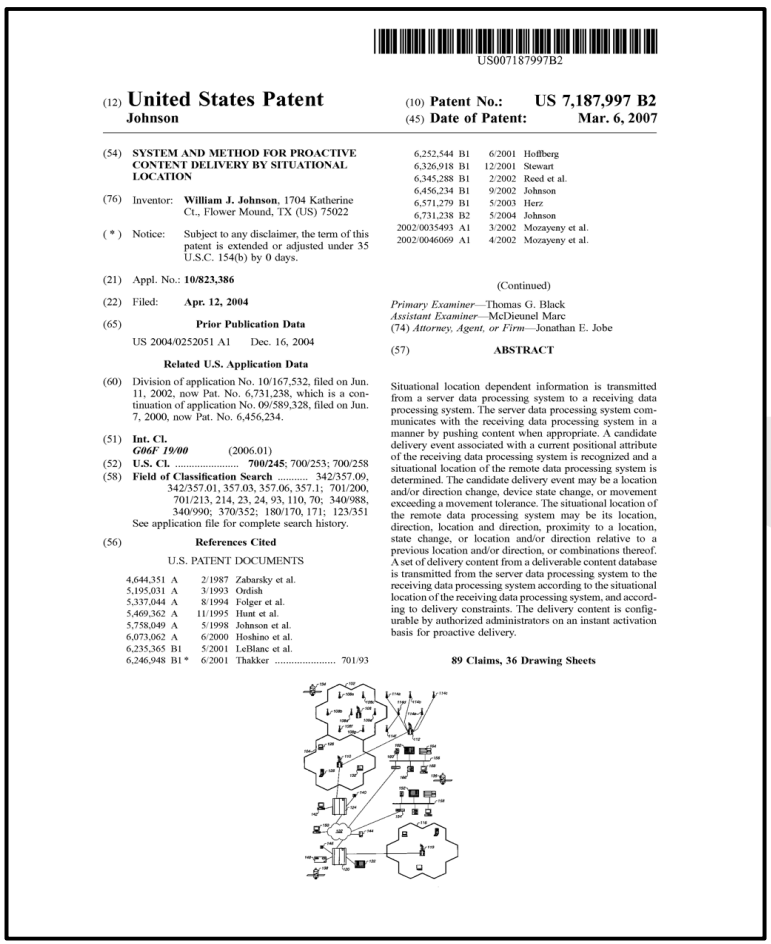
'190 patent relates to chaining images together to create a spanning image effect

Examples of Google Street View on Samsung Galaxy S



U.S. 7,187,997 (Situational Location)

- "System And Method For Proactive Content Delivery By Situational Location"
- Filing Date: April 12, 2004
  - Priority: June 7, 2000



US007187997B2

(12) **United States Patent**
Johnson

(10) **Patent No.: US 7,187,997 B2**
(45) **Date of Patent: Mar. 6, 2007**

(54) **SYSTEM AND METHOD FOR PROACTIVE CONTENT DELIVERY BY SITUATIONAL LOCATION**

(76) Inventor: **William J. Johnson**, 1704 Katherine Ct., Flower Mound, TX (US) 75022

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **10/823,386**

(22) Filed: **Apr. 12, 2004**

(65) **Prior Publication Data**

US 2004/0252051 A1 Dec. 16, 2004

Related U.S. Application Data

(60) Division of application No. 10/167,532, filed on Jun. 11, 2002, now Pat. No. 6,731,238, which is a continuation of application No. 09/589,328, filed on Jun. 7, 2000, now Pat. No. 6,456,234.

(51) **Int. Cl.**
G06F 19/00 (2006.01)

(52) **U.S. Cl.** **700/245**; 700/253; 700/258

(58) **Field of Classification Search** 342/357.09, 342/357.01, 357.03, 357.06, 357.1; 701/200, 701/213, 214, 23, 24, 93, 110, 70; 340/988, 340/990; 370/352; 180/170, 171; 123/351
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,644,351	A	2/1987	Zabarsky et al.
5,195,031	A	3/1993	Ordish
5,337,044	A	8/1994	Folger et al.
5,469,362	A	11/1995	Hunt et al.
5,758,049	A	5/1998	Johnson et al.
6,073,062	A	6/2000	Hoshino et al.
6,235,365	B1	5/2001	LeBlanc et al.
6,246,948	B1 *	6/2001	Thakker 701/93
6,252,544	B1	6/2001	Hoffberg
6,326,918	B1	12/2001	Stewart
6,345,288	B1	2/2002	Reed et al.
6,456,234	B1	9/2002	Johnson
6,571,279	B1	5/2003	Herz
6,731,238	B2	5/2004	Johnson
2002/0035493	A1	3/2002	Mozayeny et al.
2002/0046069	A1	4/2002	Mozayeny et al.

(Continued)

Primary Examiner—Thomas G. Black
Assistant Examiner—McDieunel Marc
(74) *Attorney, Agent, or Firm*—Jonathan E. Jobe

(57) **ABSTRACT**

Situational location dependent information is transmitted from a server data processing system to a receiving data processing system. The server data processing system communicates with the receiving data processing system in a manner by pushing content when appropriate. A candidate delivery event associated with a current positional attribute of the receiving data processing system is recognized and a situational location of the remote data processing system is determined. The candidate delivery event may be a location and/or direction change, device state change, or movement exceeding a movement tolerance. The situational location of the remote data processing system may be its location, direction, location and direction, proximity to a location, state change, or location and/or direction relative to a previous location and/or direction, or combinations thereof. A set of delivery content from a deliverable content database is transmitted from the server data processing system to the receiving data processing system according to the situational location of the receiving data processing system, and according to delivery constraints. The delivery content is configurable by authorized administrators on an instant activation basis for proactive delivery.

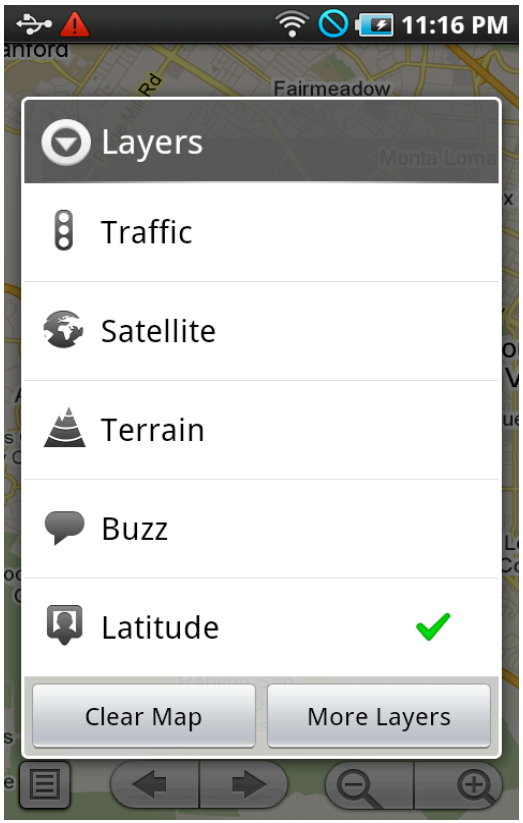
89 Claims, 36 Drawing Sheets

(57) **ABSTRACT**

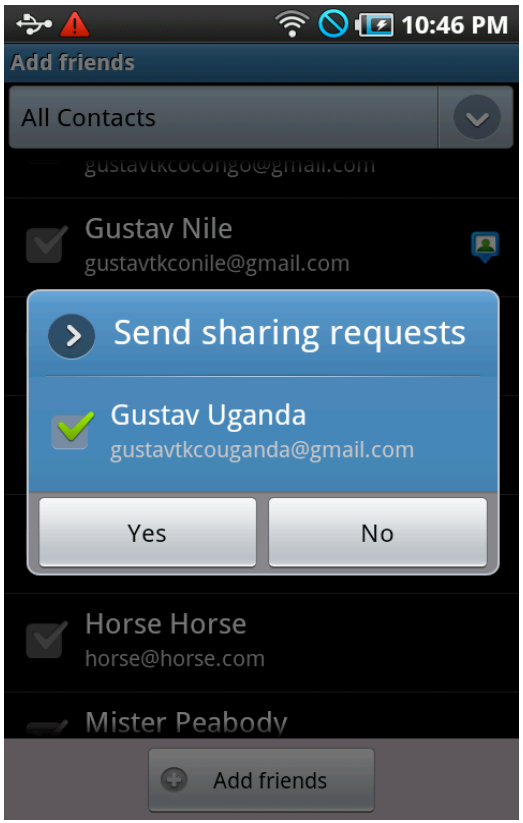
Situational location dependent information is transmitted from a server data processing system to a receiving data processing system. The server data processing system communicates with the receiving data processing system in a manner by pushing content when appropriate. A candidate delivery event associated with a current positional attribute of the receiving data processing system is recognized and a situational location of the remote data processing system is determined. The candidate delivery event may be a location and/or direction change, device state change, or movement exceeding a movement tolerance. The situational location of the remote data processing system may be its location, direction, location and direction, proximity to a location, state change, or location and/or direction relative to a previous location and/or direction, or combinations thereof. A set of delivery content from a deliverable content database is transmitted from the server data processing system to the receiving data processing system according to the situational location of the receiving data processing system, and according to delivery constraints. The delivery content is configurable by authorized administrators on an instant activation basis for proactive delivery.

U.S. 7,187,997 (Situational Location)

‘997 patent describes sending and receiving situational location dependent information to and from a mobile receiver



Google Latitude Program



Register Mobile Device on Google Latitude System

Track Physical Location